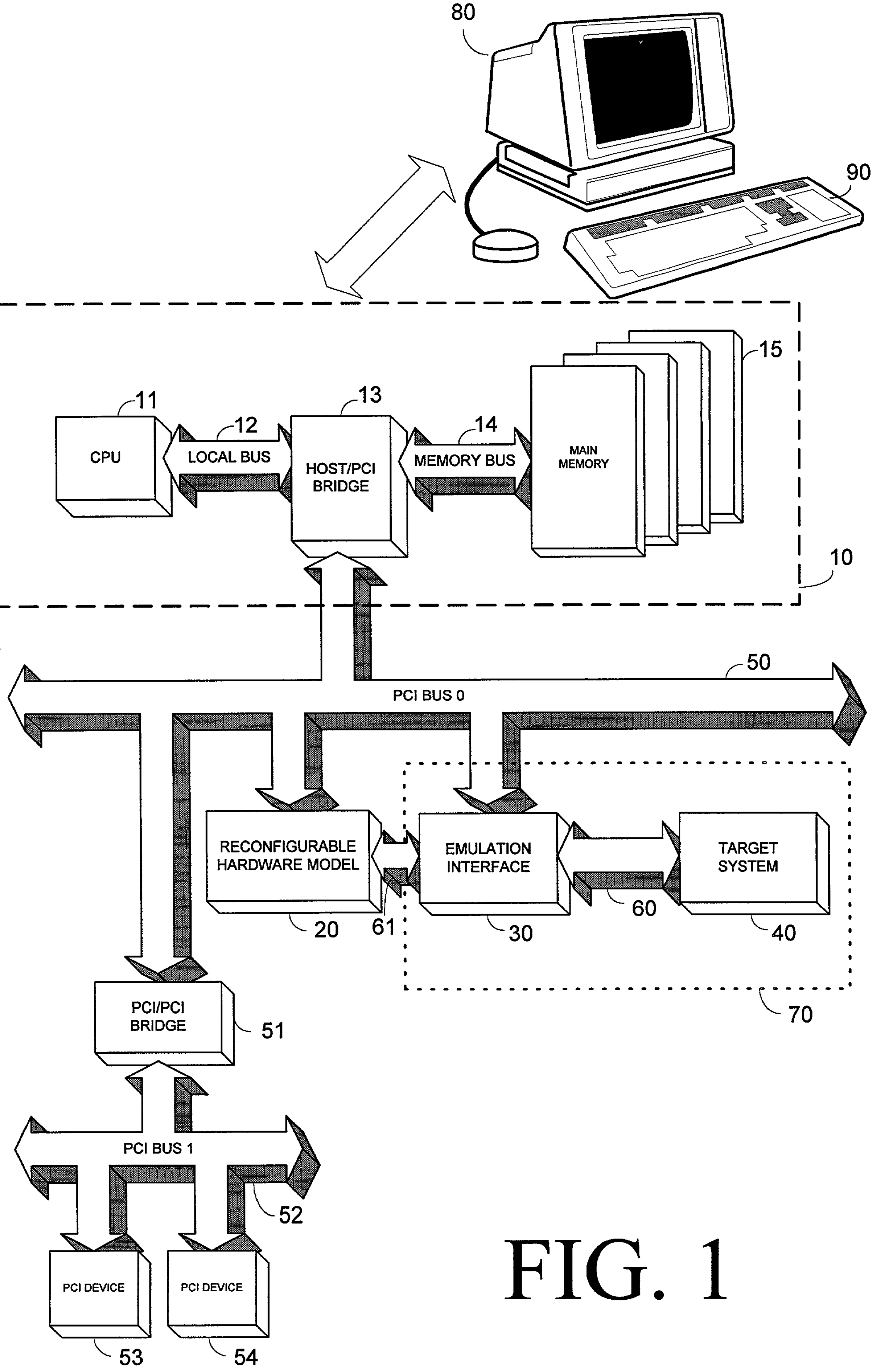

80
90
11
CPU
12
LOCAL BUS
13
HOST/PCI BRIDGE
14
MEMORY BUS
MAIN MEMORY
15
10
50
PCI BUS 0
RECONFIGURABLE HARDWARE MODEL
EMULATION INTERFACE
TARGET SYSTEM
20
61
30
60
40
70
PCI/PCI BRIDGE
51
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG. 1

USAGE

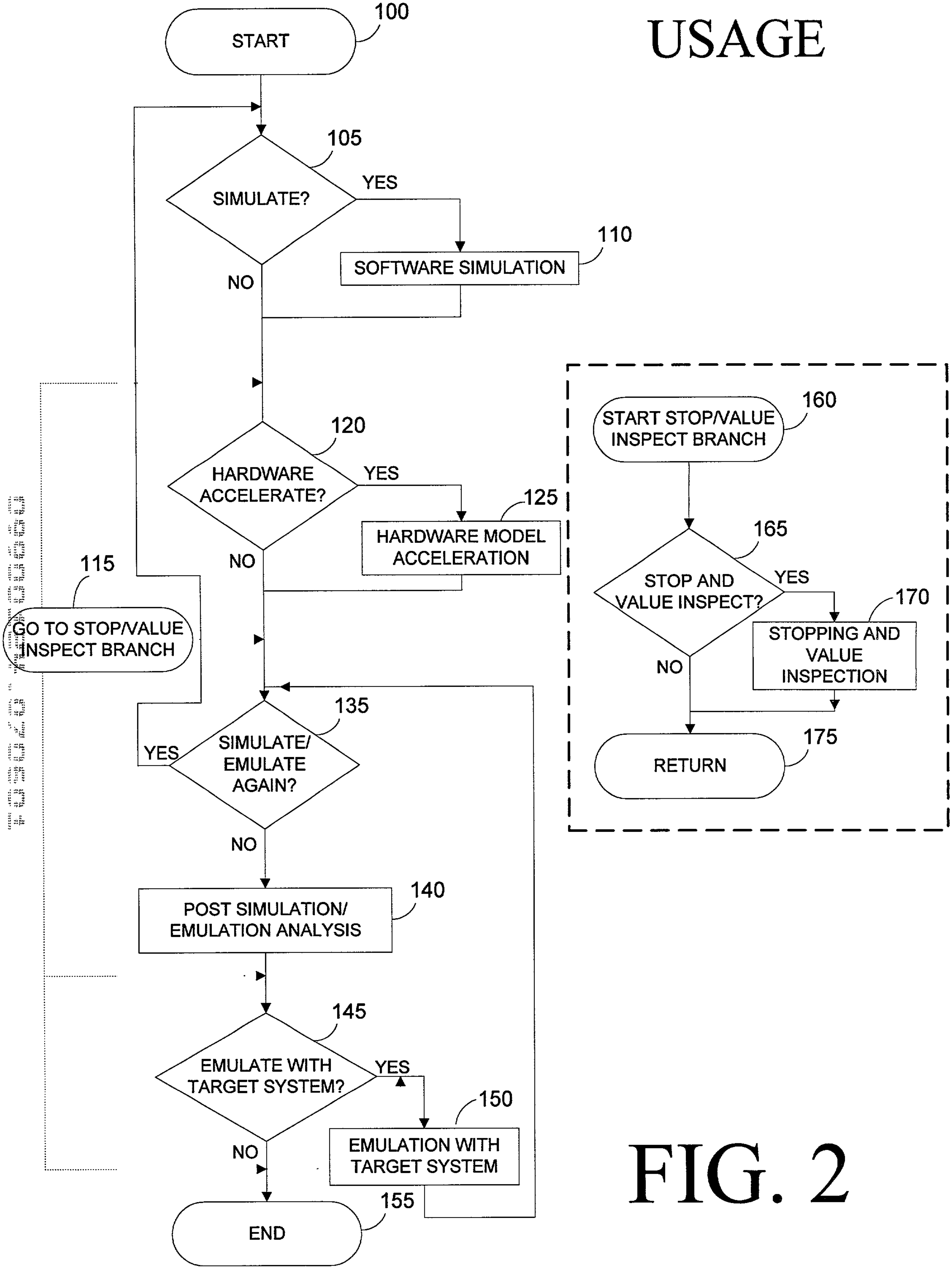

START
100
SIMULATE?
105
YES
NO
SOFTWARE SIMULATION
110
HARDWARE ACCELERATE?
120
YES
NO
HARDWARE MODEL ACCELERATION
125
115
GO TO STOP/VALUE INSPECT BRANCH
SIMULATE/ EMULATE AGAIN?
135
YES
NO
POST SIMULATION/ EMULATION ANALYSIS
140
EMULATE WITH TARGET SYSTEM?
145
YES
NO
EMULATION WITH TARGET SYSTEM
150
END
155
START STOP/VALUE INSPECT BRANCH
160
STOP AND VALUE INSPECT?
165
YES
NO
STOPPING AND VALUE INSPECTION
170
RETURN
175

FIG. 2

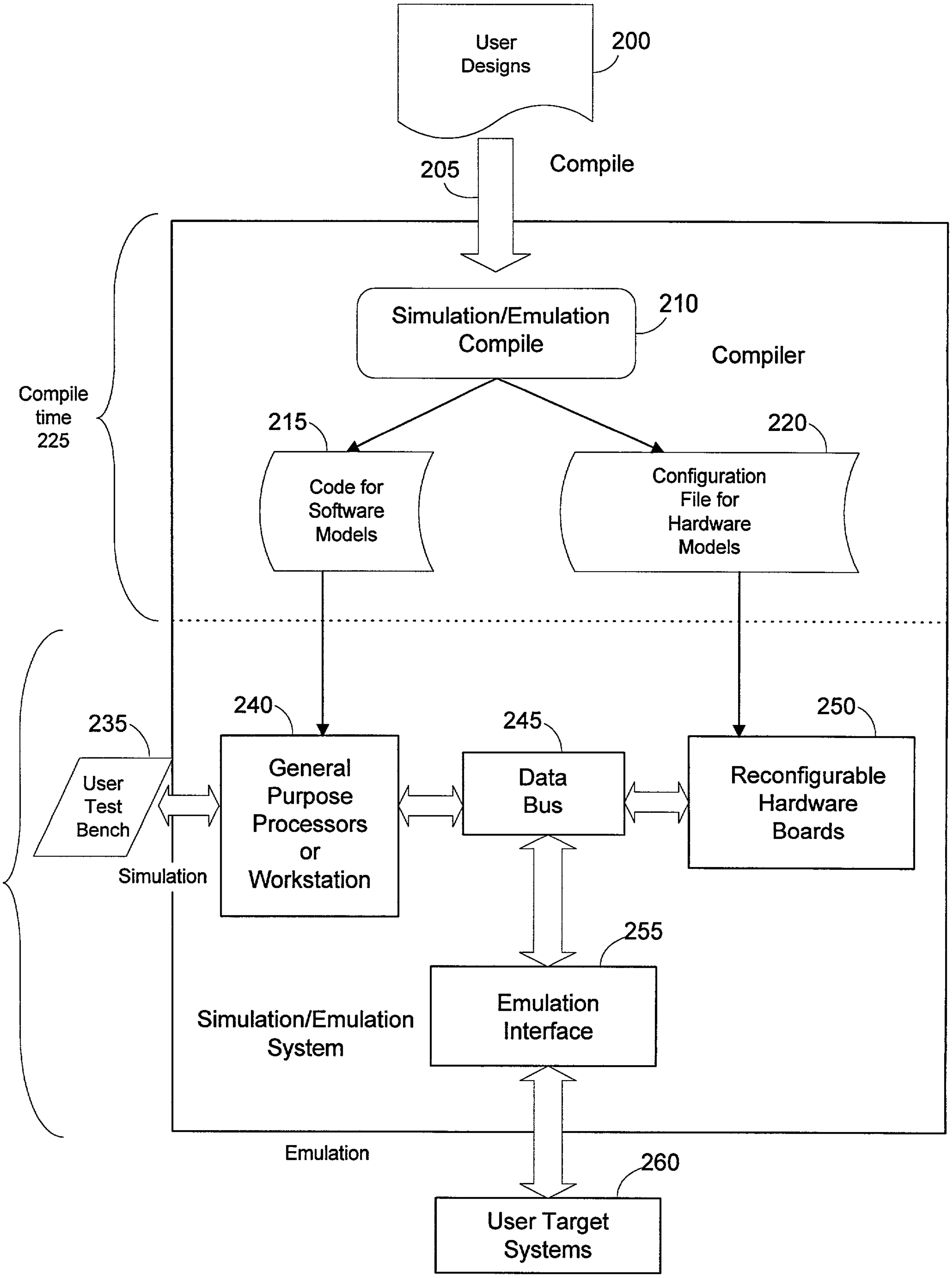
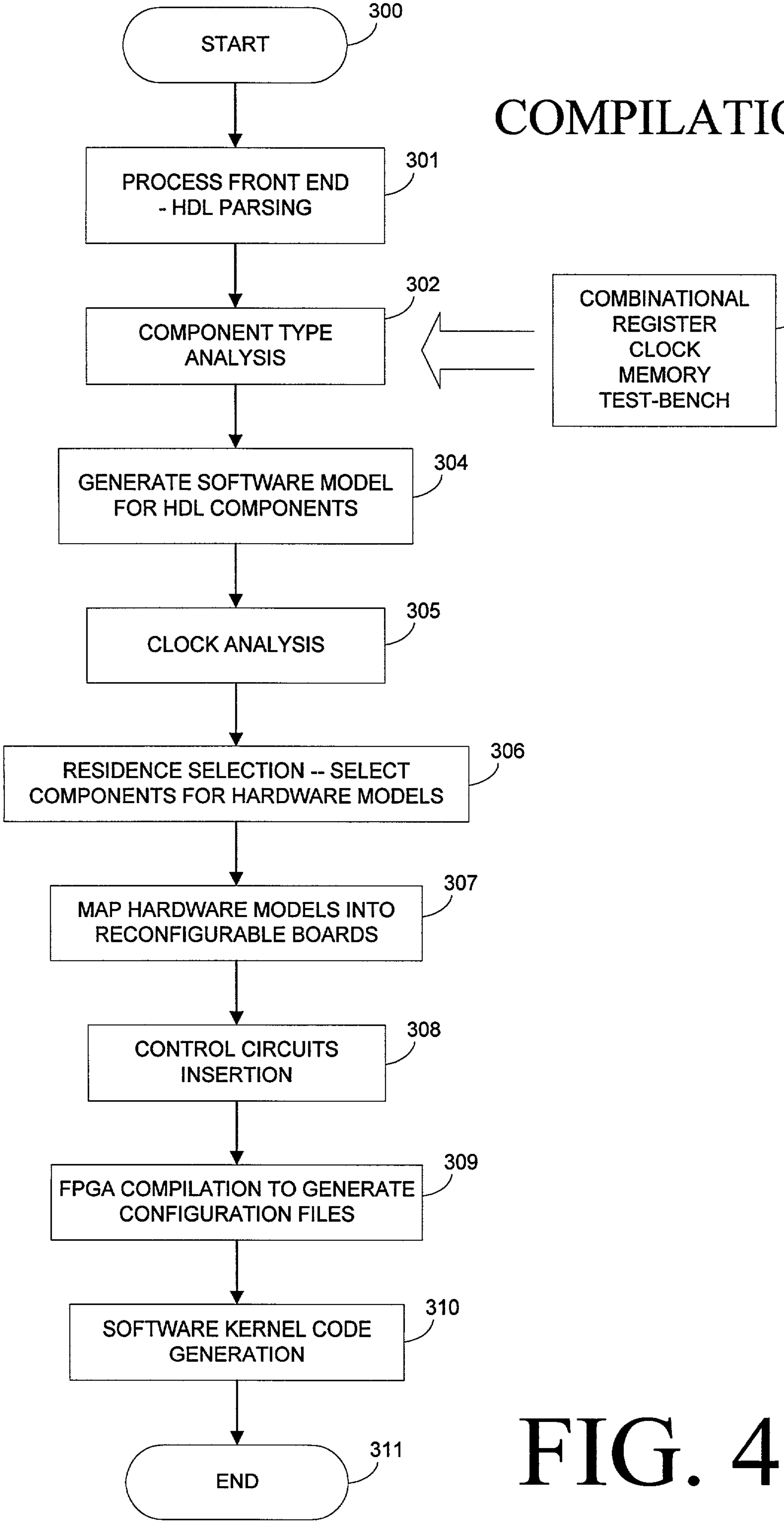
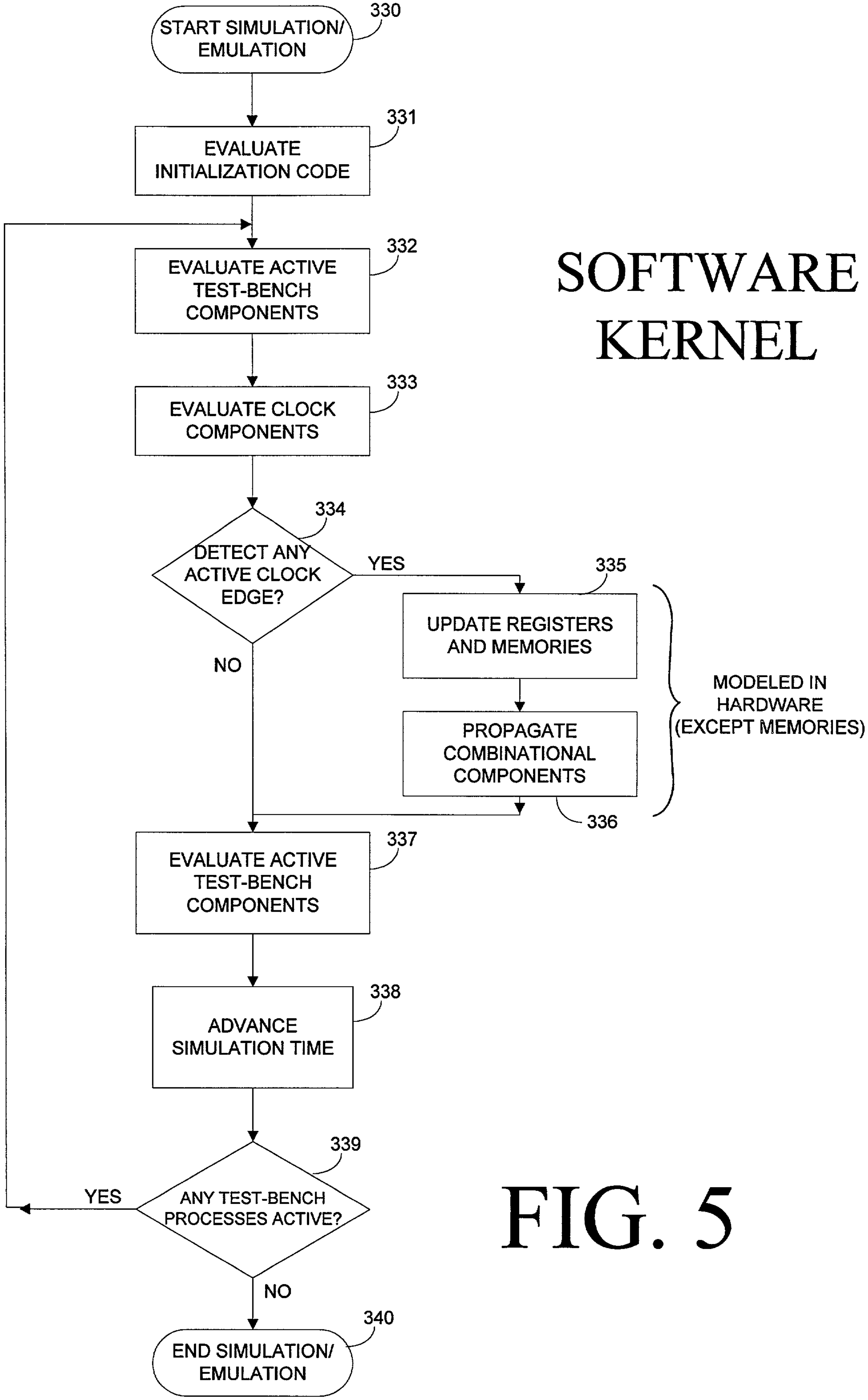

User Designs
200
205
Compile
Simulation/Emulation Compile
210
Compiler
Compile time 225
215
220
Code for Software Models
Configuration File for Hardware Models
Run time 230
235
240
245
250
User Test Bench
General Purpose Processors or Workstation
Data Bus
Reconfigurable Hardware Boards
Simulation
255
Simulation/Emulation System
Emulation Interface
Emulation
260
User Target Systems

FIG. 3


START
300
PROCESS FRONT END - HDL PARSING
301
COMPONENT TYPE ANALYSIS
302
COMBINATIONAL
REGISTER
CLOCK
MEMORY
TEST-BENCH
303
GENERATE SOFTWARE MODEL FOR HDL COMPONENTS
304
CLOCK ANALYSIS
305
RESIDENCE SELECTION -- SELECT COMPONENTS FOR HARDWARE MODELS
306
MAP HARDWARE MODELS INTO RECONFIGURABLE BOARDS
307
CONTROL CIRCUITS INSERTION
308
FPGA COMPILATION TO GENERATE CONFIGURATION FILES
309
SOFTWARE KERNEL CODE GENERATION
310
END
311

COMPILATION

FIG. 4


START SIMULATION/ EMULATION
330
EVALUATE INITIALIZATION CODE
331
EVALUATE ACTIVE TEST-BENCH COMPONENTS
332
EVALUATE CLOCK COMPONENTS
333
DETECT ANY ACTIVE CLOCK EDGE?
334
YES
NO
UPDATE REGISTERS AND MEMORIES
335
PROPAGATE COMBINATIONAL COMPONENTS
336
MODELED IN HARDWARE (EXCEPT MEMORIES)
EVALUATE ACTIVE TEST-BENCH COMPONENTS
337
ADVANCE SIMULATION TIME
338
ANY TEST-BENCH PROCESSES ACTIVE?
339
YES
NO
END SIMULATION/ EMULATION
340

SOFTWARE KERNEL

FIG. 5

# MAPPING HARDWARE MODELS TO RECONFIGURABLE BOARDS

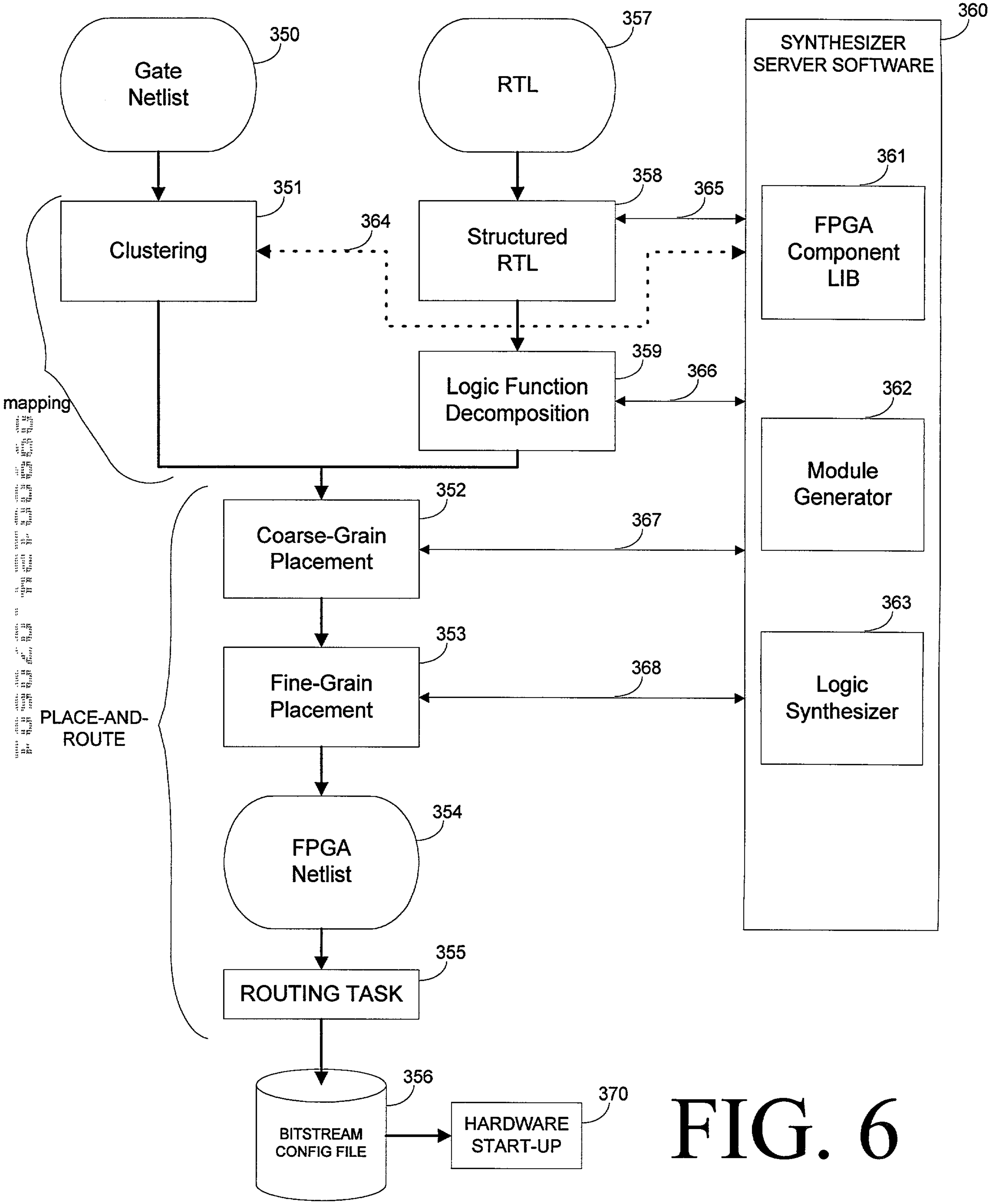
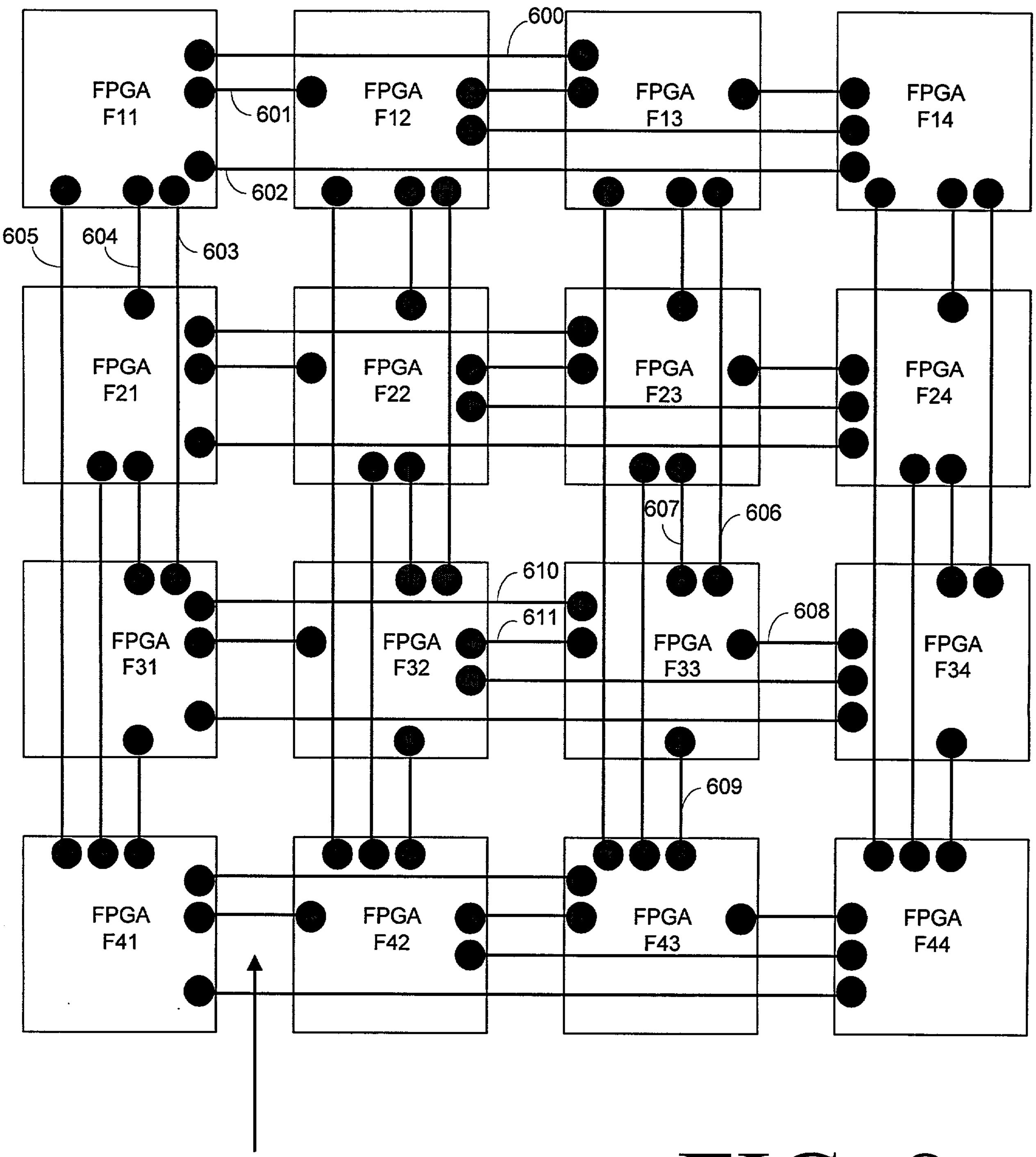


FIG. 6

|  | F11 | F12 | F13 | F14 | F21 | F22 | F23 | F24 | F31 | F32 | F33 | F34 | F41 | F42 | F43 | F44 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F11 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| F12 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| F13 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| F14 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| F21 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| F22 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| F23 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| F24 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| F31 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| F32 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| F33 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| F34 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| F41 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| F42 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| F43 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| F44 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

FIG. 7

FPGA INTERCONNECTION



1/6 of total I/O pins of FPGA for interconnection

FIG. 8

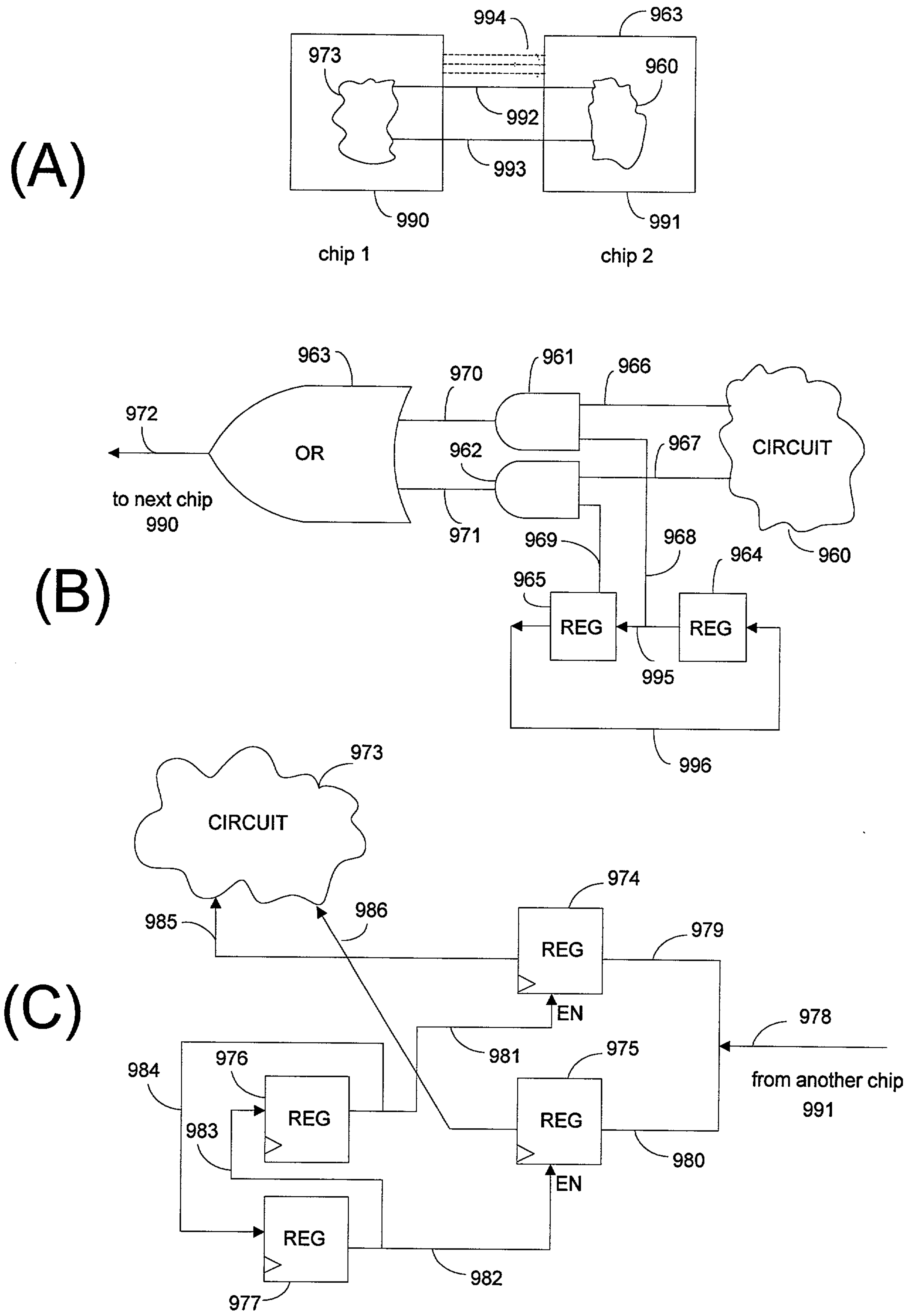

(A)
994
963
973
960
992
993
990
991
chip 1
chip 2
(B)
963
961
966
970
972
OR
962
967
CIRCUIT
to next chip
990
971
969
968
964
960
965
REG
REG
995
996
(C)
973
CIRCUIT
974
985
986
REG
979
EN
978
981
975
from another chip
991
984
976
REG
983
REG
980
EN
REG
982
977

FIG. 9

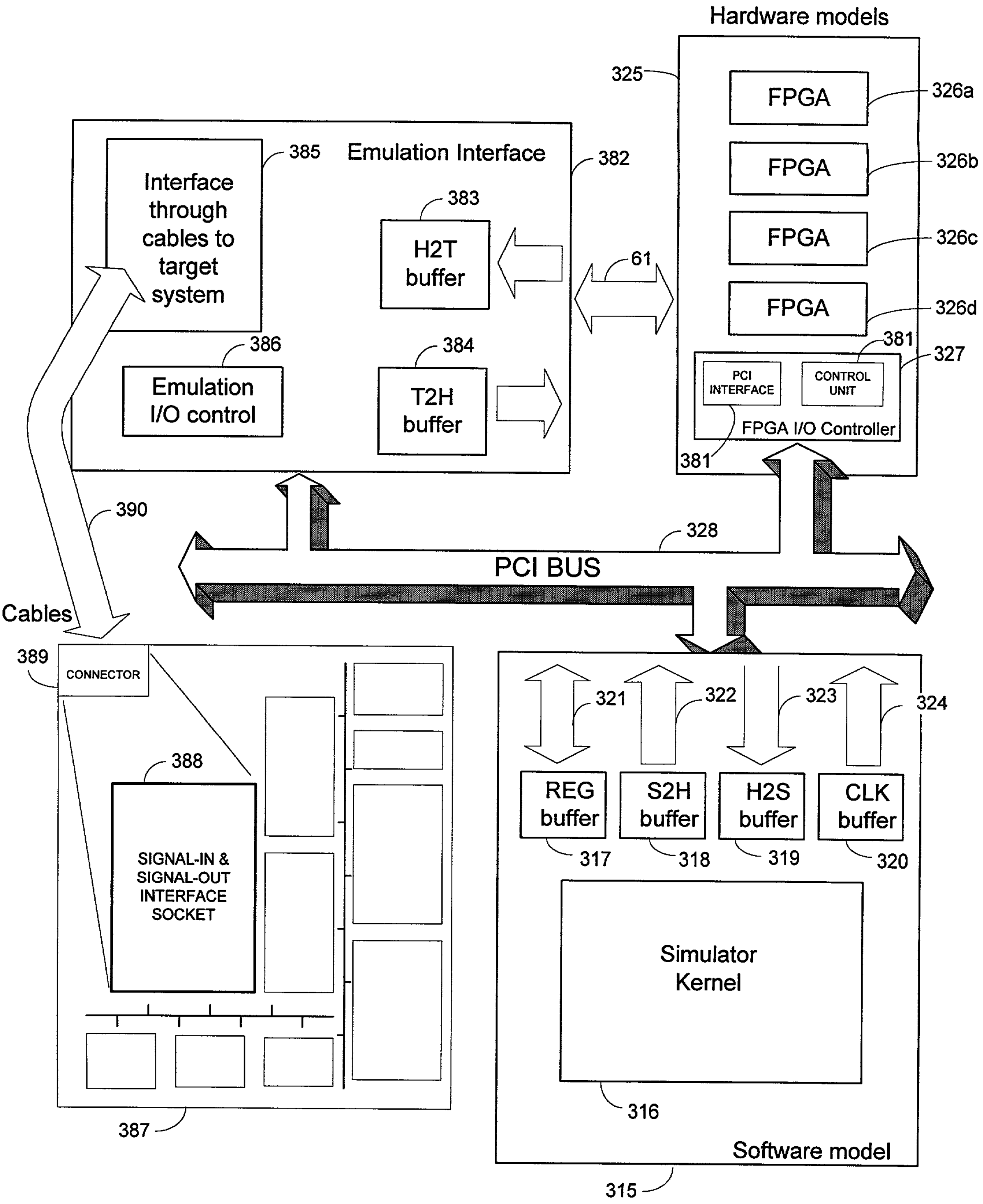

Hardware models
325
FPGA
326a
FPGA
326b
FPGA
326c
FPGA
326d
381
327
PCI INTERFACE
CONTROL UNIT
FPGA I/O Controller
381
385
Emulation Interface
382
Interface through cables to target system
383
H2T buffer
61
386
Emulation I/O control
384
T2H buffer
390
328
PCI BUS
Cables
389
CONNECTOR
388
SIGNAL-IN & SIGNAL-OUT INTERFACE SOCKET
387
321
322
323
324
REG buffer
S2H buffer
H2S buffer
CLK buffer
317
318
319
320
Simulator Kernel
316
Software model
315

FIG. 10

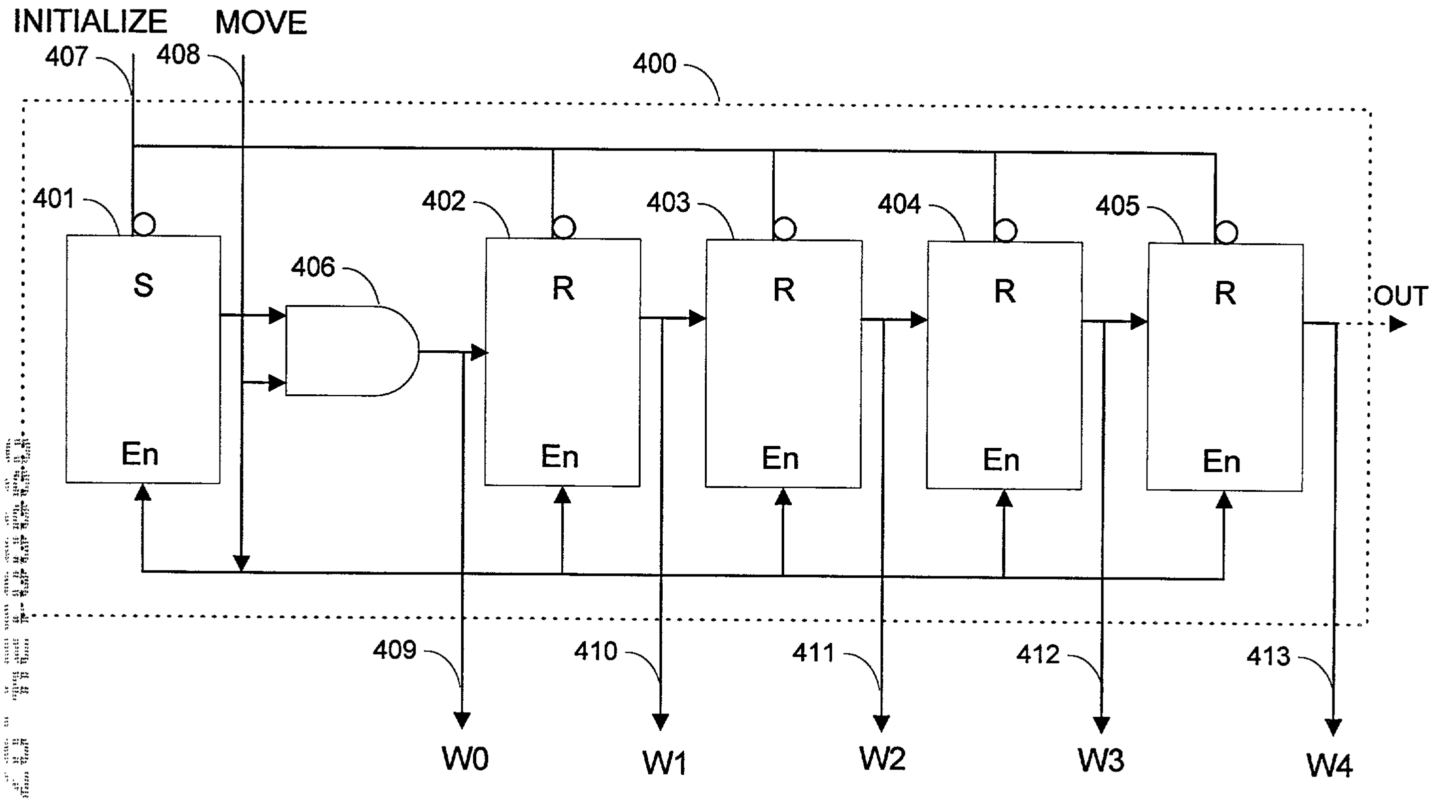

INITIALIZE
MOVE
407
408
400
401
402
403
404
405
406
S
R
R
R
R
OUT
En
En
En
En
En
409
410
411
412
413
W0
W1
W2
W3
W4

FIG. 11

# ADDRESS POINTER INITIALIZATION

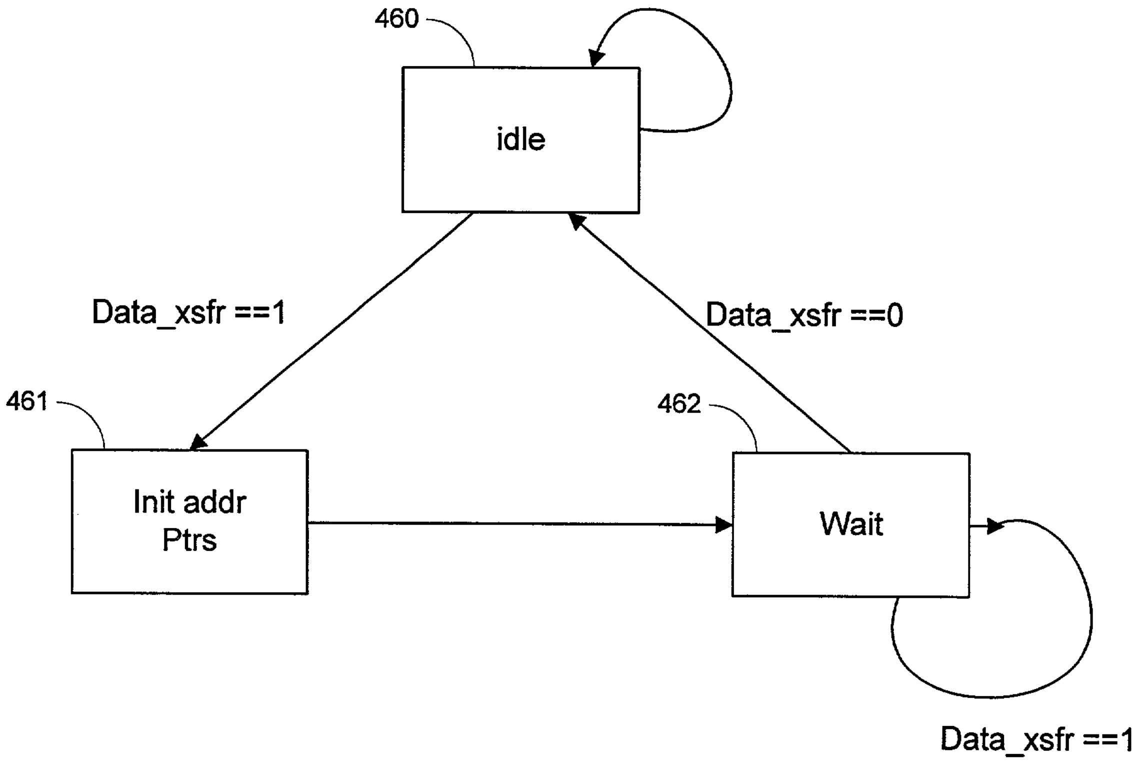


FIG. 12

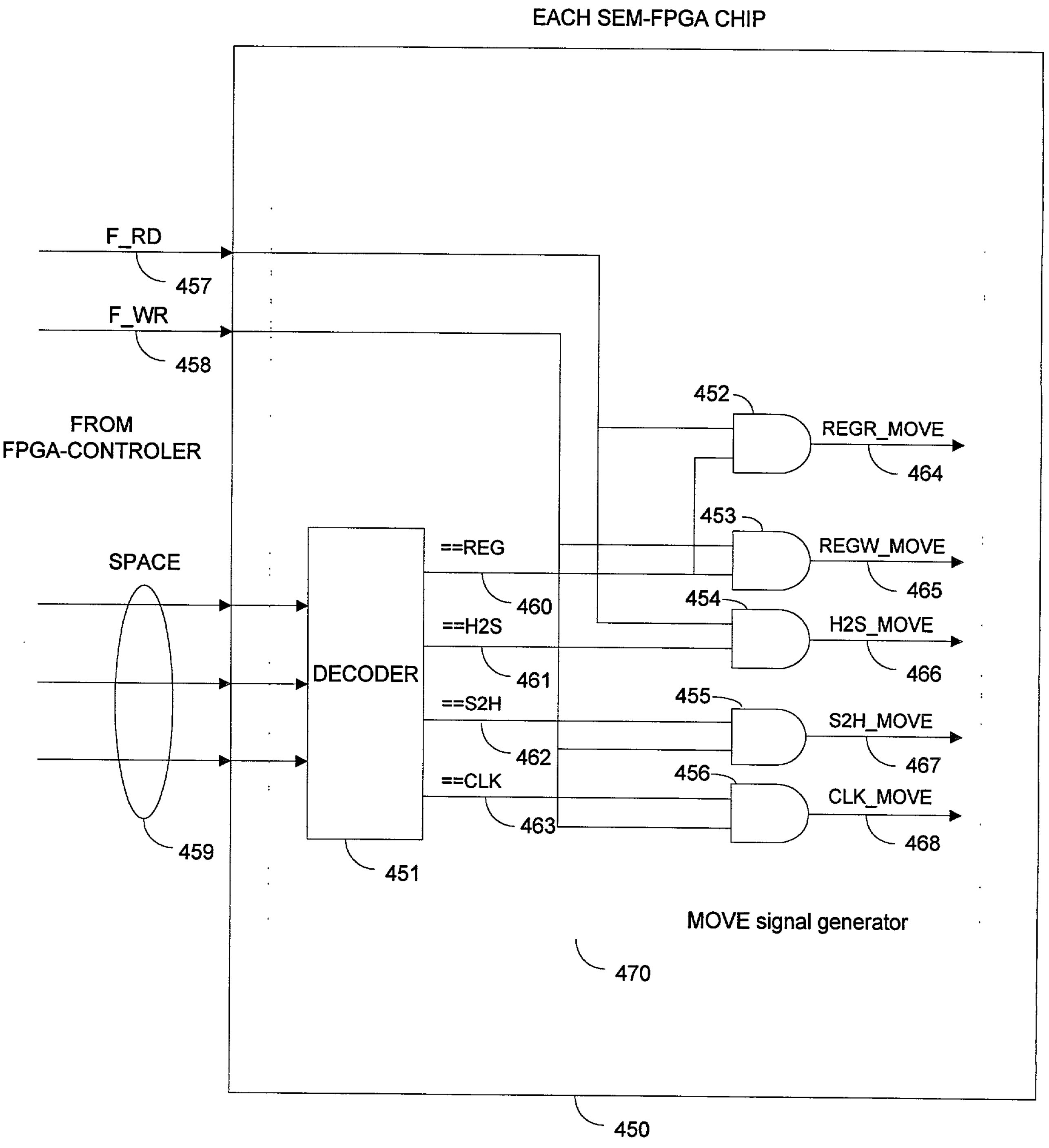

EACH SEM-FPGA CHIP
F_RD
457
F_WR
458
FROM
FPGA-CONTROLER
SPACE
459
DECODER
451
==REG
460
==H2S
461
==S2H
462
==CLK
463
452
REGR_MOVE
464
453
REGW_MOVE
465
454
H2S_MOVE
466
455
S2H_MOVE
467
456
CLK_MOVE
468
MOVE signal generator
470
450

FIG. 13

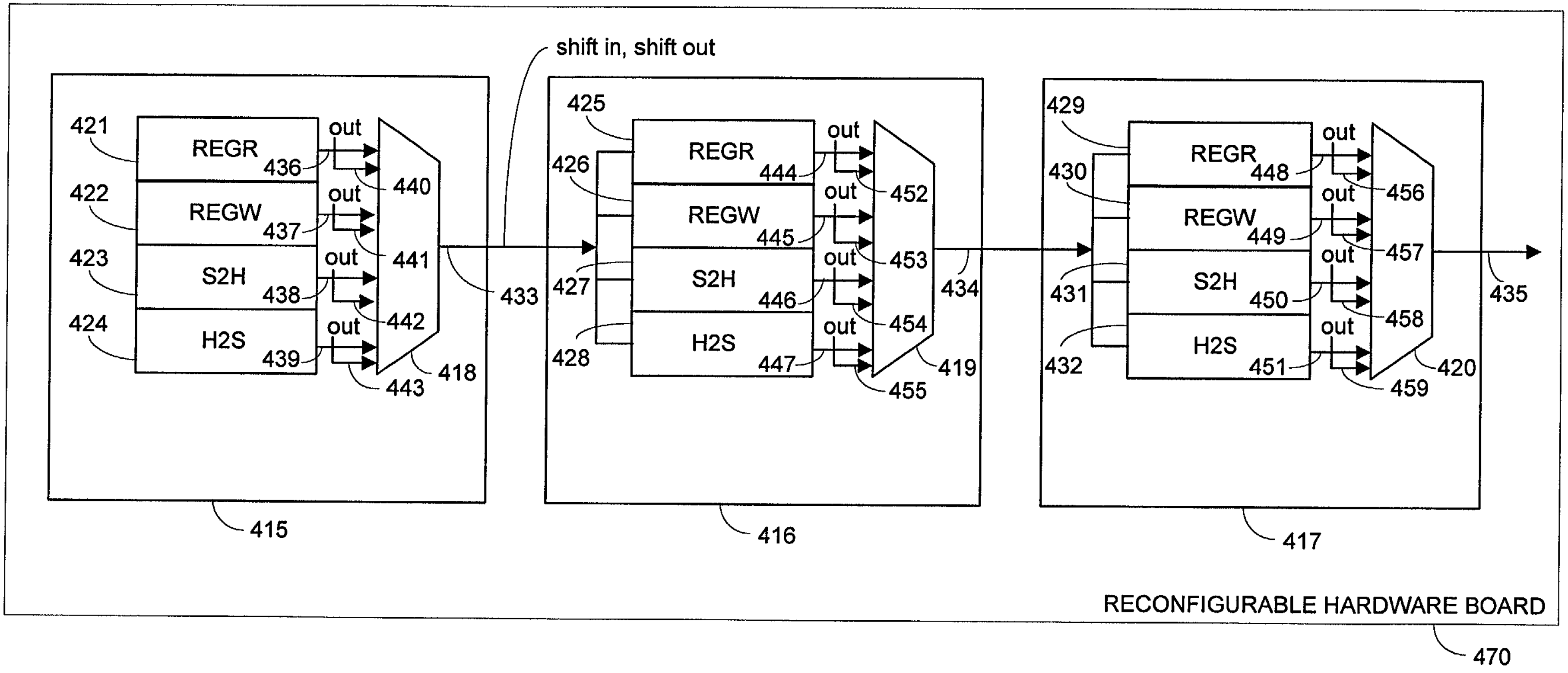

shift in, shift out
421
422
423
424
REGR
REGW
S2H
H2S
436
437
438
439
out
out
out
out
440
441
442
443
418
433
415
425
426
427
428
REGR
REGW
S2H
H2S
444
445
446
447
out
out
out
out
452
453
454
455
419
434
416
429
430
431
432
REGR
REGW
S2H
H2S
448
449
450
451
out
out
out
out
456
457
458
459
420
435
417
RECONFIGURABLE HARDWARE BOARD
470

FIG. 14

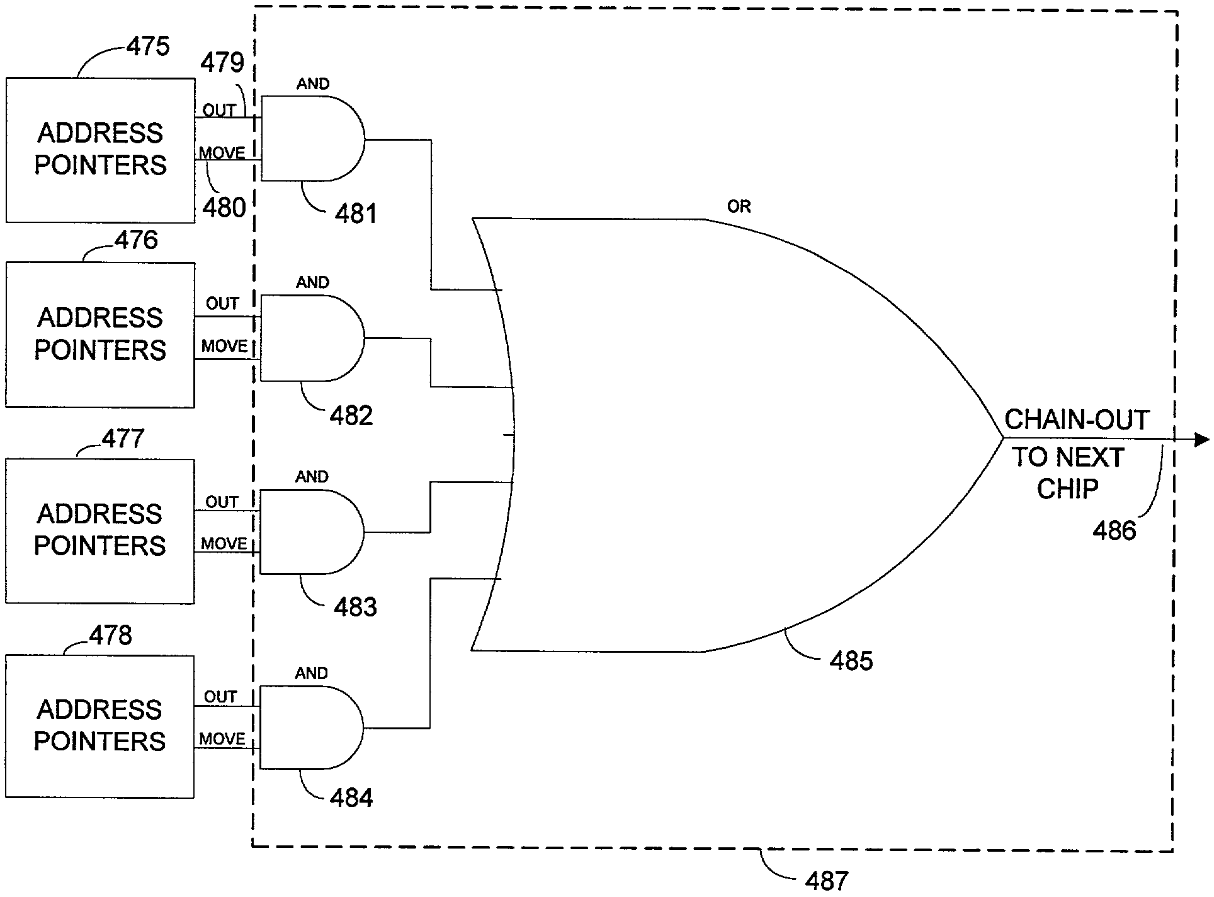

475
ADDRESS POINTERS
476
ADDRESS POINTERS
477
ADDRESS POINTERS
478
ADDRESS POINTERS
479
OUT
MOVE
480
OUT
MOVE
OUT
MOVE
OUT
MOVE
AND
481
AND
482
AND
483
AND
484
OR
485
CHAIN-OUT TO NEXT CHIP
486
487

FIG. 15

## GATED DATA/CLOCK ANALYSIS

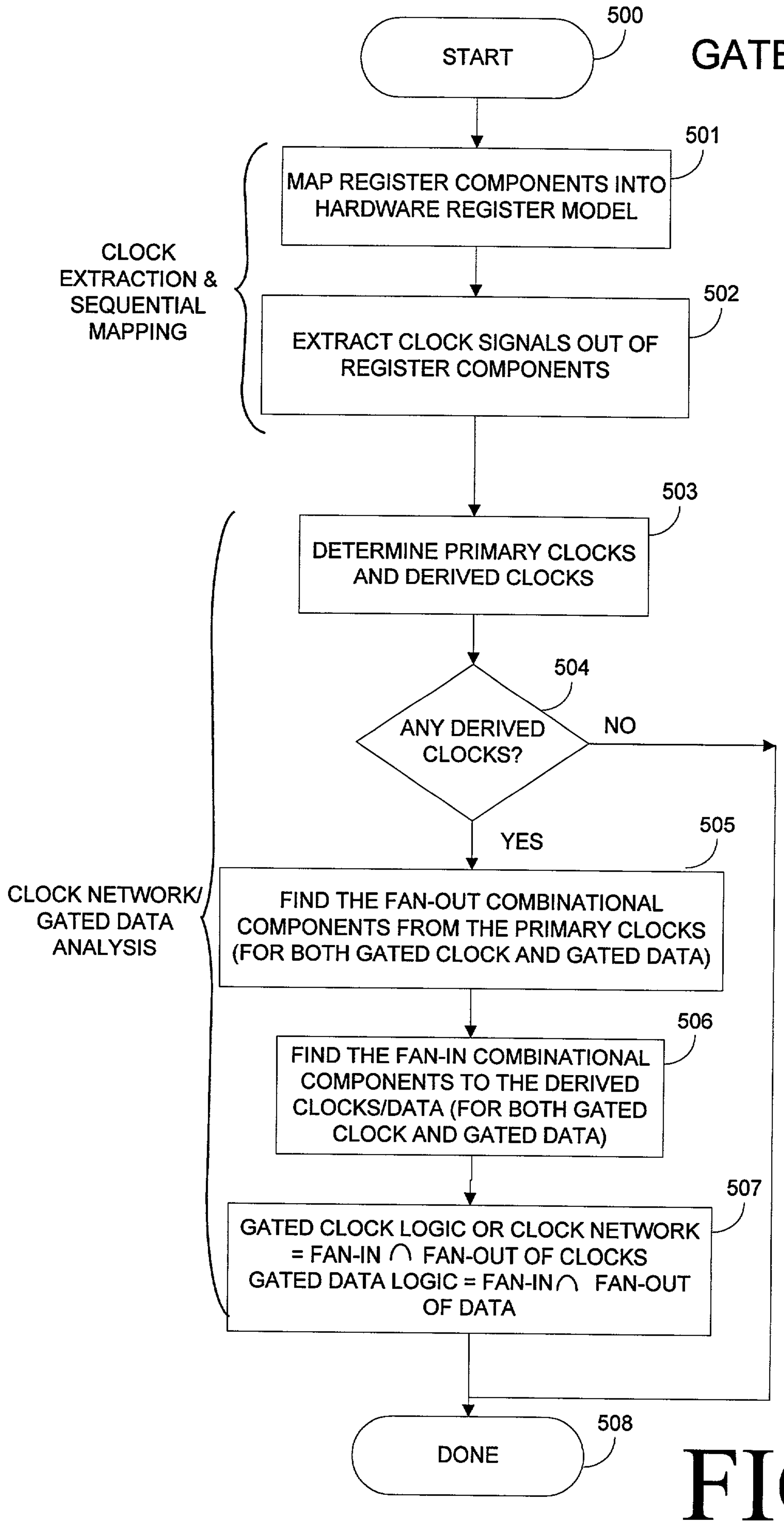

500
START
501
MAP REGISTER COMPONENTS INTO HARDWARE REGISTER MODEL
CLOCK EXTRACTION & SEQUENTIAL MAPPING
502
EXTRACT CLOCK SIGNALS OUT OF REGISTER COMPONENTS
503
DETERMINE PRIMARY CLOCKS AND DERIVED CLOCKS
504
ANY DERIVED CLOCKS?
NO
YES
505
FIND THE FAN-OUT COMBINATIONAL COMPONENTS FROM THE PRIMARY CLOCKS (FOR BOTH GATED CLOCK AND GATED DATA)
CLOCK NETWORK/ GATED DATA ANALYSIS
506
FIND THE FAN-IN COMBINATIONAL COMPONENTS TO THE DERIVED CLOCKS/DATA (FOR BOTH GATED CLOCK AND GATED DATA)
507
GATED CLOCK LOGIC OR CLOCK NETWORK = FAN-IN ∩ FAN-OUT OF CLOCKS
GATED DATA LOGIC = FAN-IN ∩ FAN-OUT OF DATA
508
DONE

FIG. 16

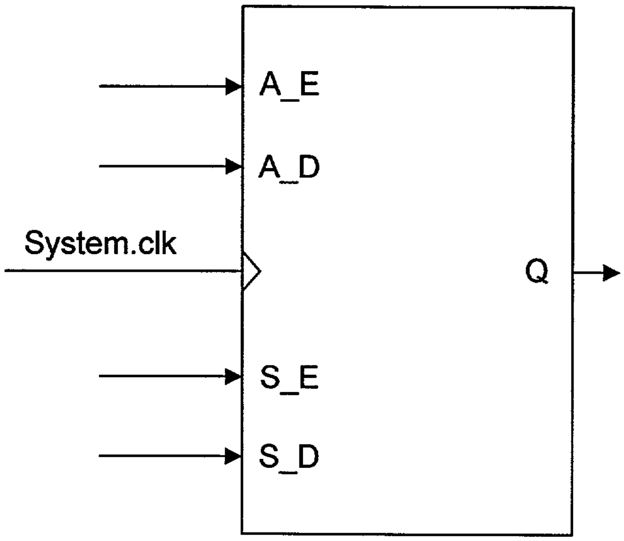


FIG. 17

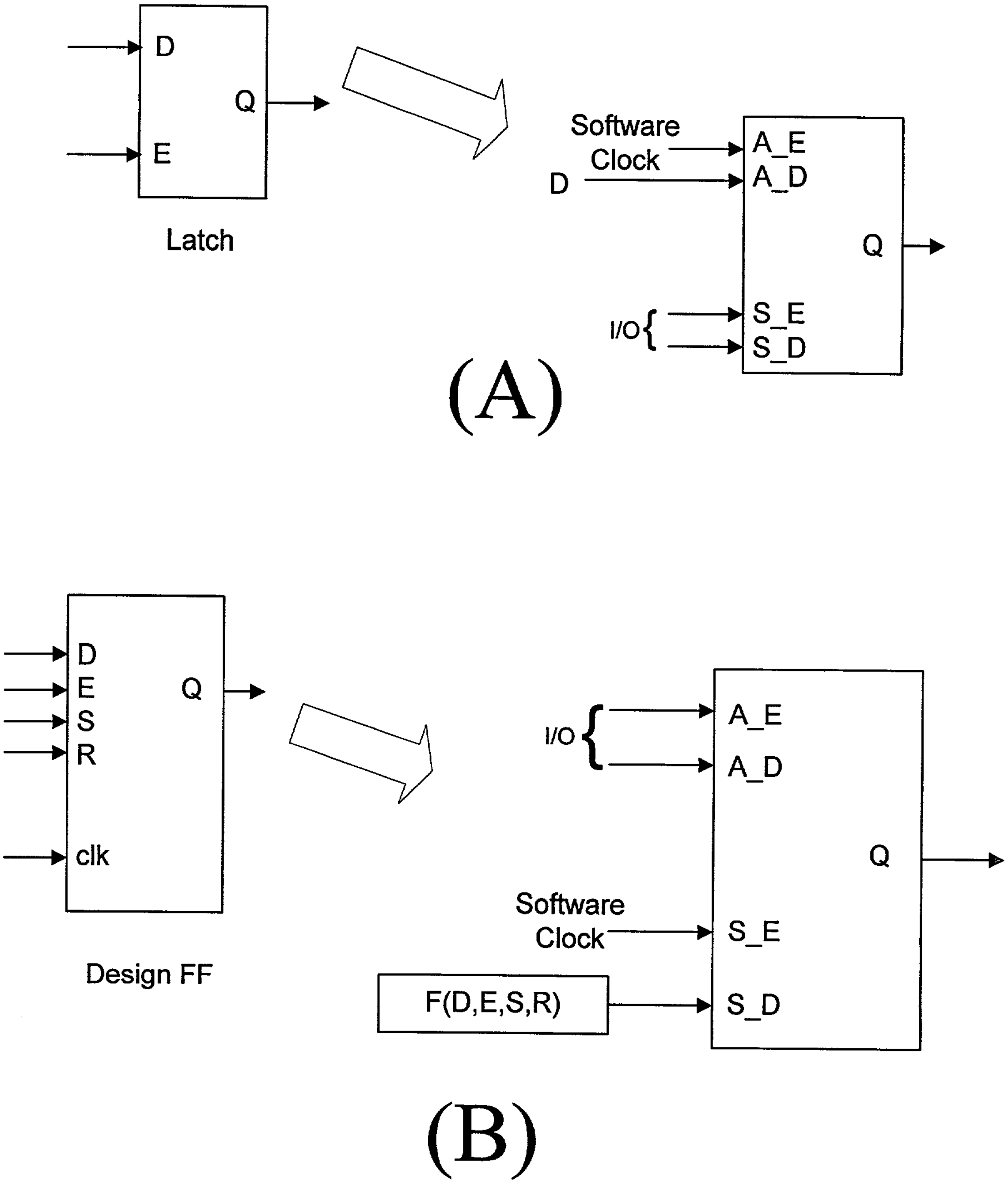

D
E
Q
Latch
Software
Clock
D
A_E
A_D
Q
I/O
S_E
S_D
(A)
D
E
S
R
clk
Q
Design FF
I/O
A_E
A_D
Q
Software
Clock
S_E
F(D,E,S,R)
S_D
(B)

FIG. 18

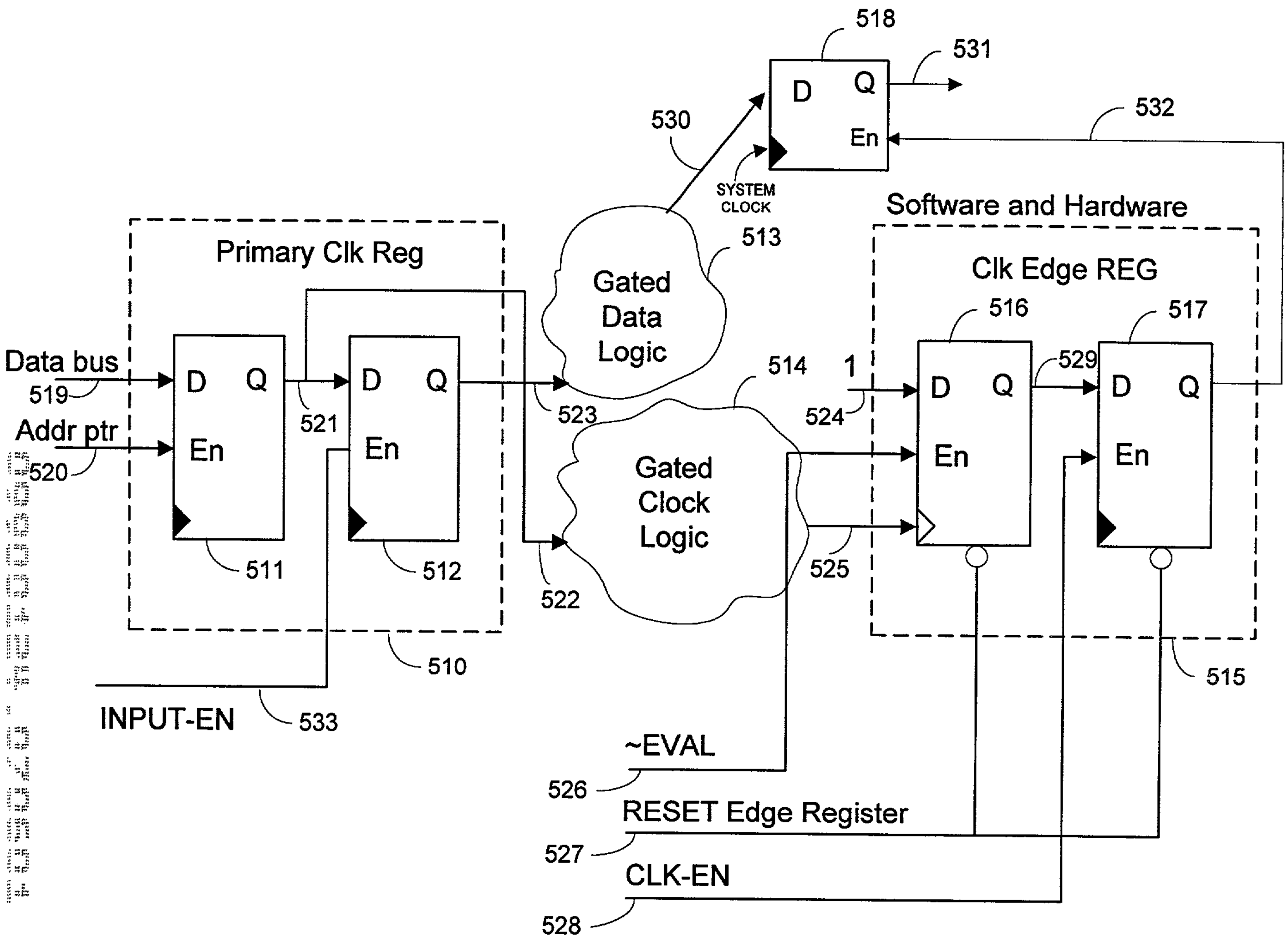

518
531
D
Q
En
530
SYSTEM CLOCK
532
Software and Hardware
513
Primary Clk Reg
Gated Data Logic
Clk Edge REG
516
517
Data bus
519
D
Q
D
Q
523
514
1
D
Q
529
D
Q
524
Addr ptr
521
En
En
En
En
520
Gated Clock Logic
511
512
522
525
510
515
INPUT-EN
533
~EVAL
526
RESET Edge Register
527
CLK-EN
528

FIG. 19

DURING
EVALUATION

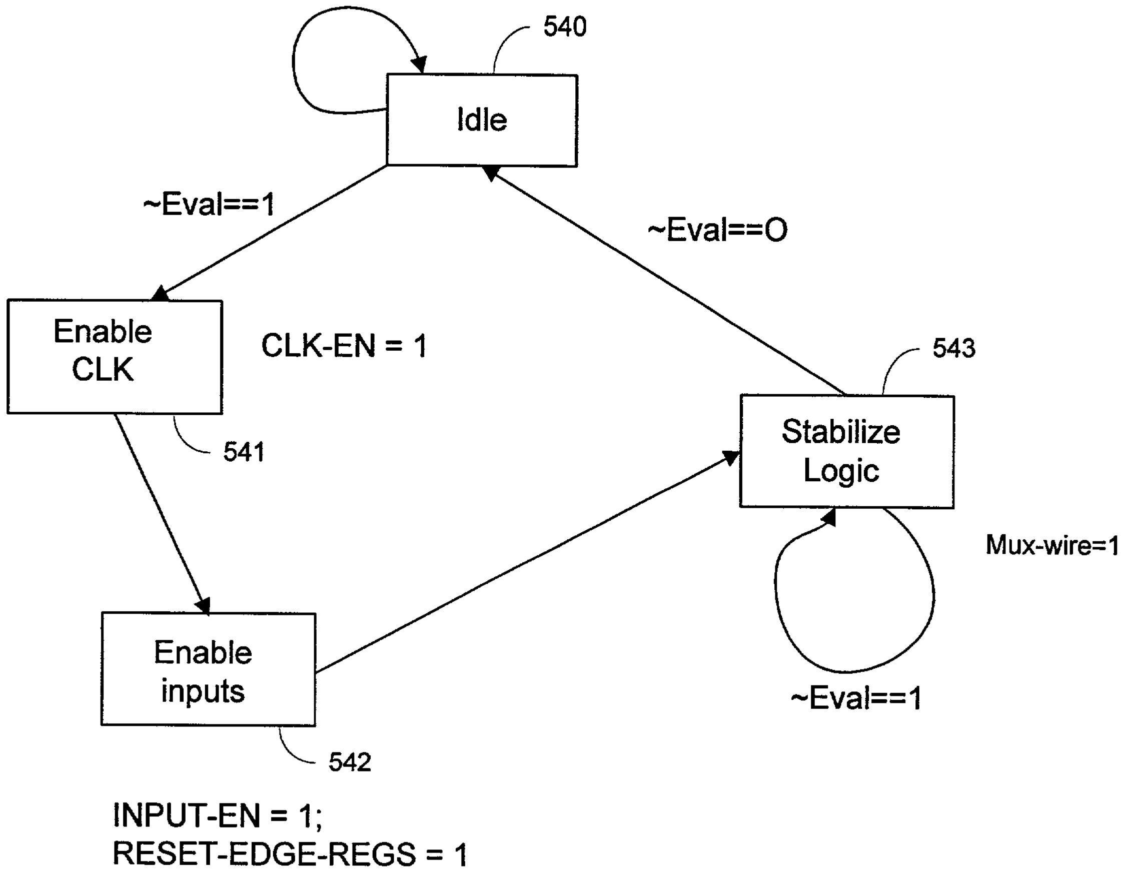


FIG. 20

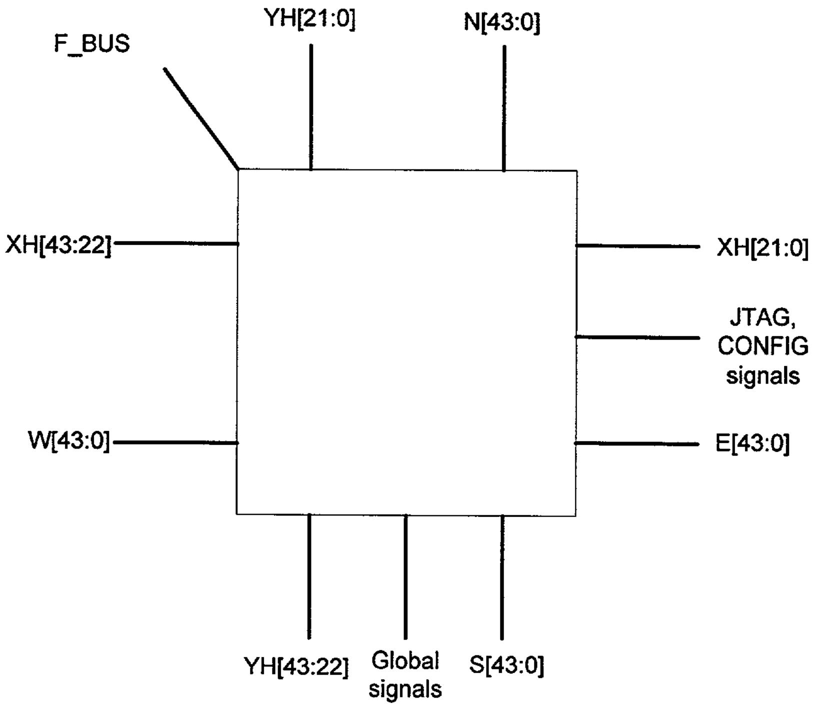
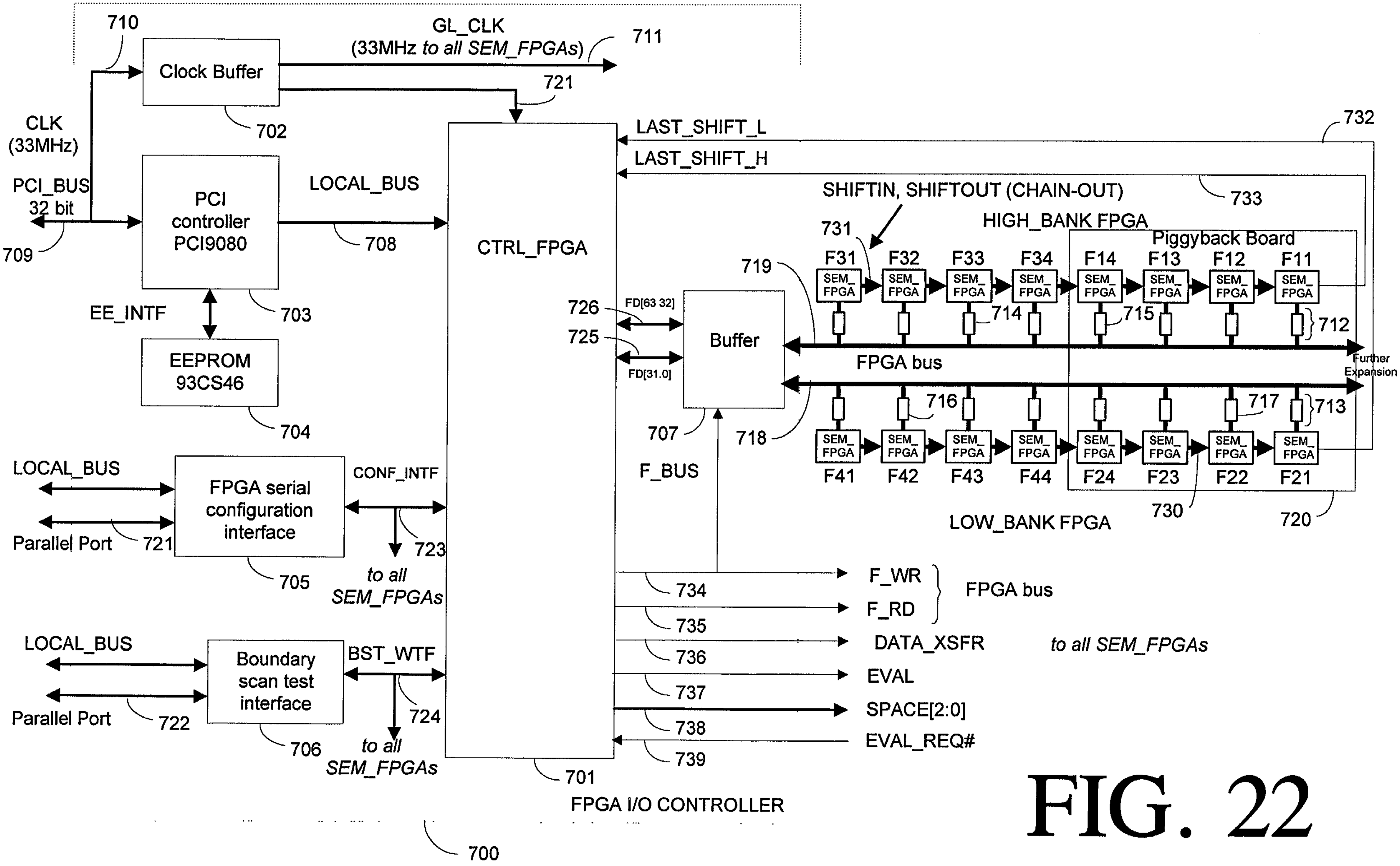

F_BUS
YH[21:0]
N[43:0]
XH[43:22]
XH[21:0]
JTAG,
CONFIG
signals
W[43:0]
E[43:0]
YH[43:22]
Global
signals
S[43:0]

FIG. 21



710
Clock Buffer
702
GL_CLK
(33MHz *to all SEM_FPGAs*)
711
721
CLK
(33MHz)
PCI_BUS
32 bit
709
PCI
controller
PCI9080
703
LOCAL_BUS
708
EE_INTF
EEPROM
93CS46
704
CTRL_FPGA
LAST_SHIFT_L
LAST_SHIFT_H
732
733
SHIFTIN, SHIFTOUT (CHAIN-OUT)
731
HIGH_BANK FPGA
Piggyback Board
F31
F32
F33
F34
F14
F13
F12
F11
SEM_
FPGA
714
715
712
726
FD[63 32]
725
FD[31.0]
Buffer
719
FPGA bus
Further
Expansion
716
717
713
718
707
F_BUS
F41
F42
F43
F44
F24
F23
F22
F21
730
720
LOW_BANK FPGA
LOCAL_BUS
FPGA serial
configuration
interface
Parallel Port
721
705
CONF_INTF
723
*to all*
*SEM_FPGAs*
LOCAL_BUS
Boundary
scan test
interface
Parallel Port
722
706
BST_WTF
724
*to all*
*SEM_FPGAs*
734
F_WR
F_RD
FPGA bus
735
DATA_XSFR
*to all SEM_FPGAs*
736
EVAL
737
SPACE[2:0]
738
EVAL_REQ#
739
701
FPGA I/O CONTROLLER
700

FIG. 22

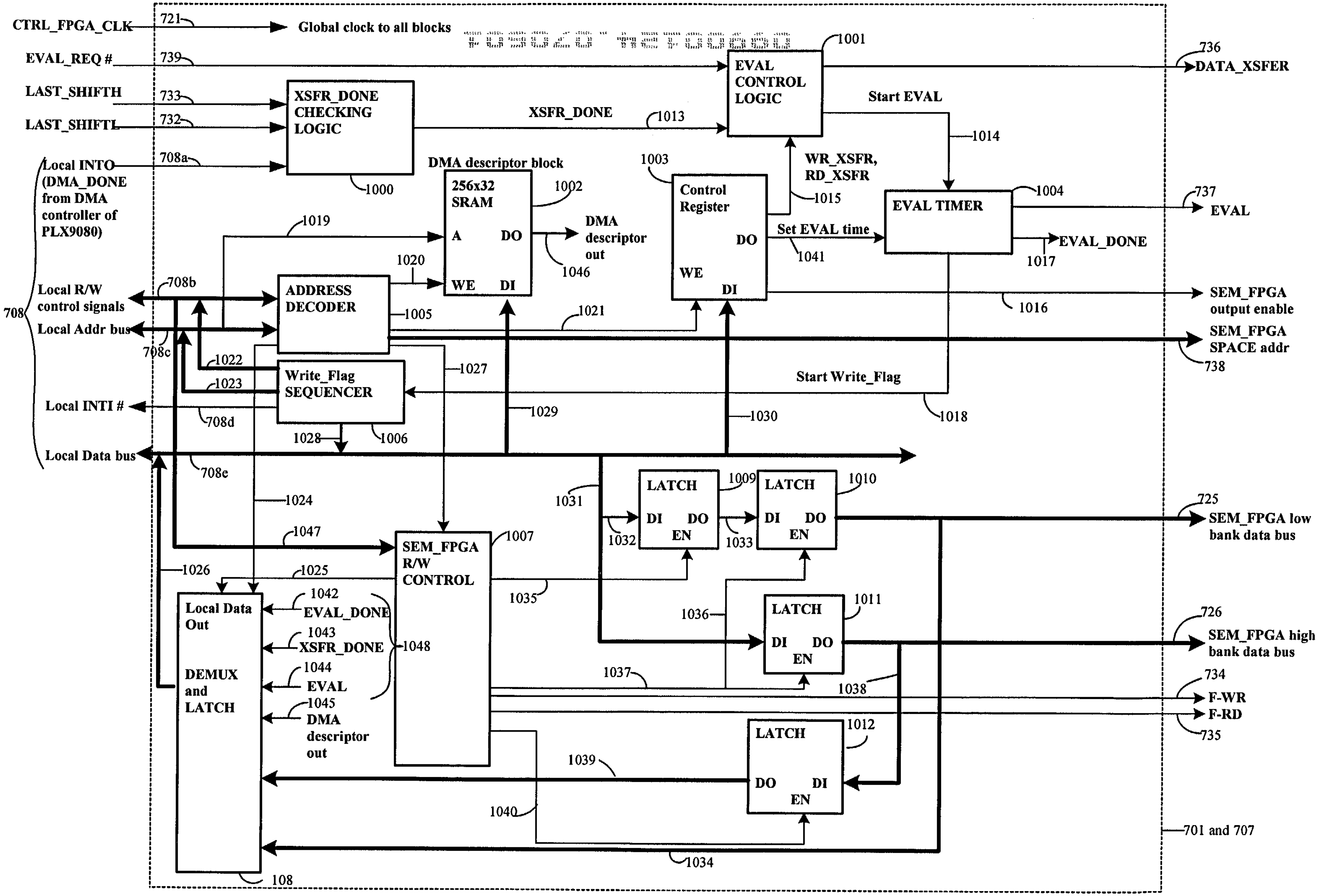

CTRL_FPGA_CLK
721
Global clock to all blocks
EVAL_REQ #
739
LAST_SHIFTH
733
LAST_SHIFTL
732
Local INTO (DMA_DONE from DMA controller of PLX9080)
708a
708
Local R/W control signals
708b
Local Addr bus
708c
Local INTI #
708d
Local Data bus
708e
XSFR_DONE CHECKING LOGIC
1000
XSFR_DONE
1013
EVAL CONTROL LOGIC
1001
736
DATA_XSFER
Start EVAL
1014
DMA descriptor block
256x32 SRAM
A
DO
WE
DI
1002
DMA descriptor out
1046
1019
1020
ADDRESS DECODER
1005
1021
1022
1023
Write_Flag SEQUENCER
1006
1027
1028
1029
1003
Control Register
DO
WE
DI
WR_XSFR, RD_XSFR
1015
Set EVAL time
1041
EVAL TIMER
1004
737
EVAL
EVAL_DONE
1017
SEM_FPGA output enable
1016
SEM_FPGA SPACE addr
738
Start Write_Flag
1018
1030
1024
1031
LATCH
DI
DO
EN
1009
1032
1033
LATCH
DI
DO
EN
1010
725
SEM_FPGA low bank data bus
1047
SEM_FPGA R/W CONTROL
1007
1048
1026
1025
1035
Local Data Out
DEMUX and LATCH
108
1042
EVAL_DONE
1043
XSFR_DONE
1044
EVAL
1045
DMA descriptor out
1036
LATCH
DI
DO
EN
1011
726
SEM_FPGA high bank data bus
1037
1038
734
F-WR
F-RD
735
LATCH
DO
DI
EN
1012
1039
1040
1034
701 and 707

FIG. 23

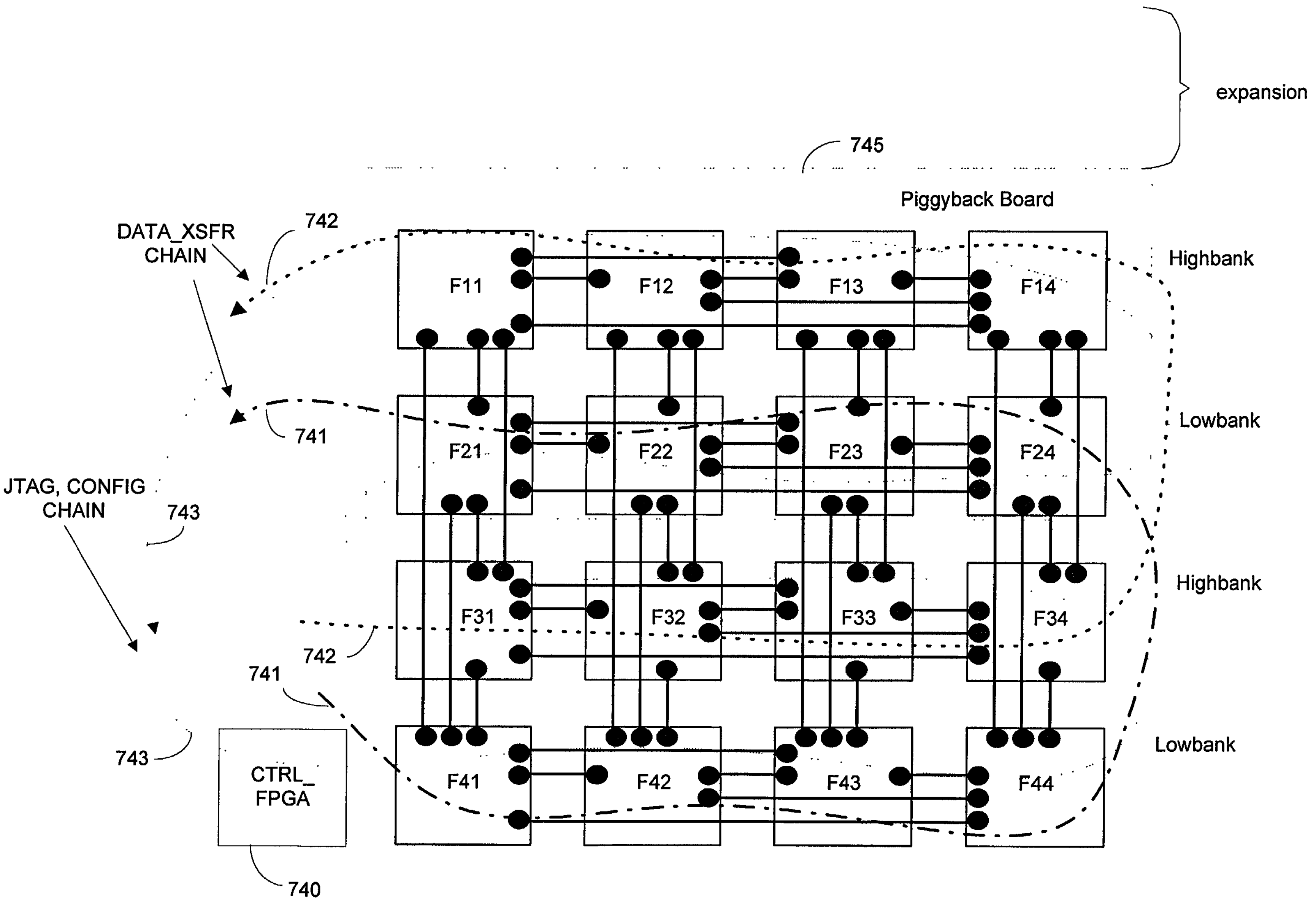
expansion
745
Piggyback Board
DATA_XSFR
CHAIN
742
741
JTAG, CONFIG
CHAIN
743
742
741
743
CTRL_
FPGA
740
F11
F12
F13
F14
F21
F22
F23
F24
F31
F32
F33
F34
F41
F42
F43
F44
Highbank
Lowbank
Highbank
Lowbank

FIG. 24

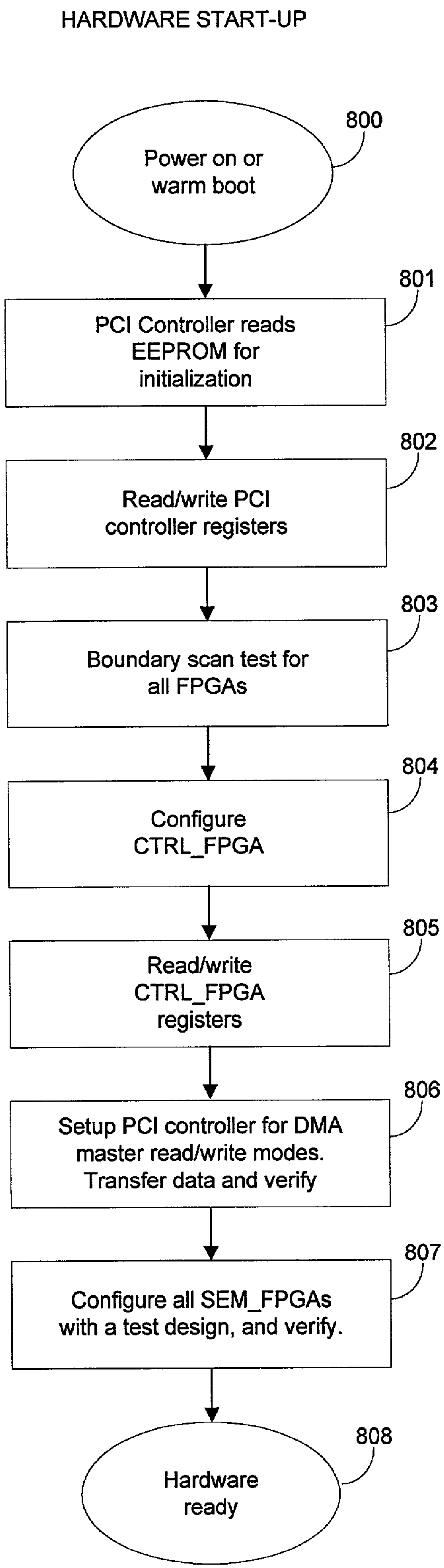

HARDWARE START-UP
800
Power on or warm boot
801
PCI Controller reads EEPROM for initialization
802
Read/write PCI controller registers
803
Boundary scan test for all FPGAs
804
Configure CTRL_FPGA
805
Read/write CTRL_FPGA registers
806
Setup PCI controller for DMA master read/write modes. Transfer data and verify
807
Configure all SEM_FPGAs with a test design, and verify.
808
Hardware ready

FIG. 25

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(posedge clock or negedge reset)
    if(□reset)
        q = 0;
    else
        q = d;

endmodule

module example;
    wire d1, d2, d3;
    wire q1, q2, q3;

    reg sigin;
    wire sigout;
    reg clk, reset;

    register reg1 (clk, reset, d1, q1);
    register reg2 (clk, reset, d2, q2);
    register reg3 (clk, reset, d3, q3);

    assign d1 = sigin ^ q3;
    assign d2 = q1 ^ q3;
    assign d3 = q2 ^ q3;
    assign sigout = q3;

    // a clock generator
    always
    begin
        clk = 0;
        #5;
        clk = 1;
        #5;
    end

    // a signal generator
    always
    begin
        #10;
        sigin = $random;
    end

    // initialization
    initial
    begin
        reset = 0;
        sigin = 0;
        #1;
        reset =1;
        #5;
        $monitor($time, " %b, %b", sigin, sigout);
        #1000 $finish;
    end
    end module
```

FIG. 26

CIRCUIT DIAGRAM

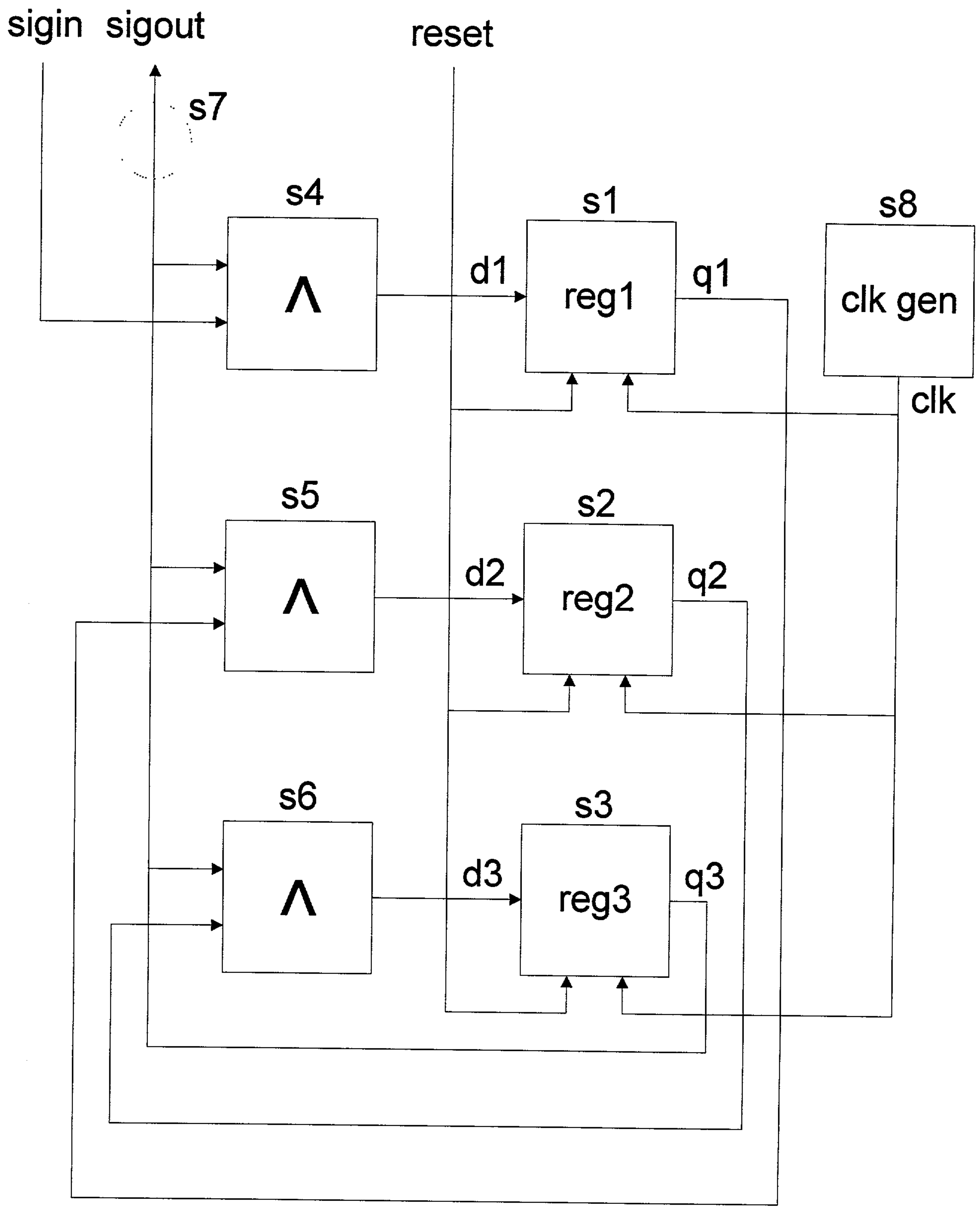
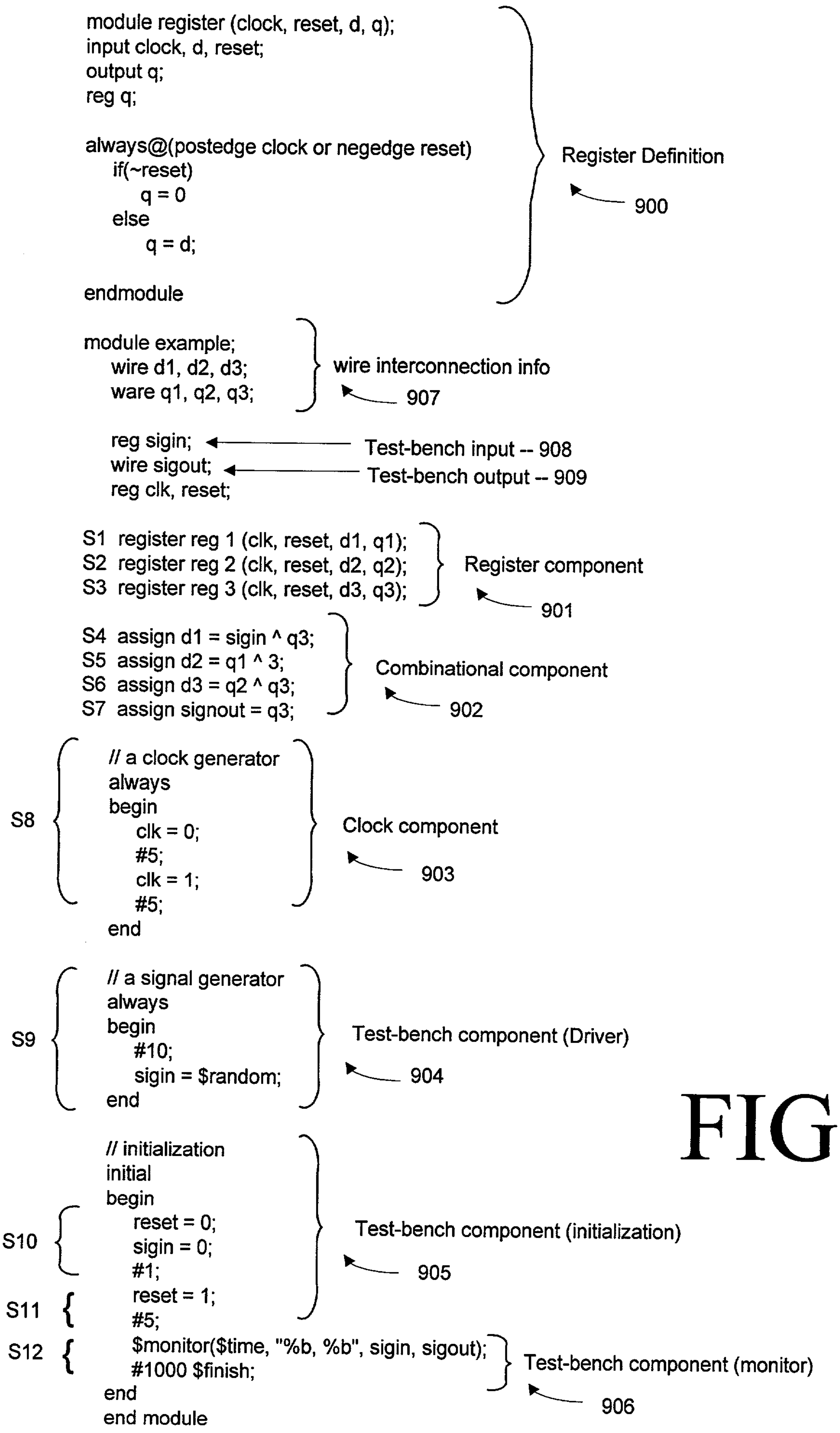


FIG. 27



```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(postedge clock or negedge reset)
    if(~reset)
        q = 0
    else
        q = d;

endmodule

module example;
    wire d1, d2, d3;
    ware q1, q2, q3;

    reg sigin;
    wire sigout;
    reg clk, reset;

S1  register reg 1 (clk, reset, d1, q1);
S2  register reg 2 (clk, reset, d2, q2);
S3  register reg 3 (clk, reset, d3, q3);

S4  assign d1 = sigin ^ q3;
S5  assign d2 = q1 ^ 3;
S6  assign d3 = q2 ^ q3;
S7  assign signout = q3;

        // a clock generator
        always
        begin
S8          clk = 0;
            #5;
            clk = 1;
            #5;
        end

        // a signal generator
        always
        begin
S9          #10;
            sigin = $random;
        end

        // initialization
        initial
        begin
S10         reset = 0;
            sigin = 0;
            #1;
S11         reset = 1;
            #5;
S12         $monitor($time, "%b, %b", sigin, sigout);
            #1000 $finish;
        end
        end module
```

FIG. 28

SIGNAL NETWORK ANALYSIS

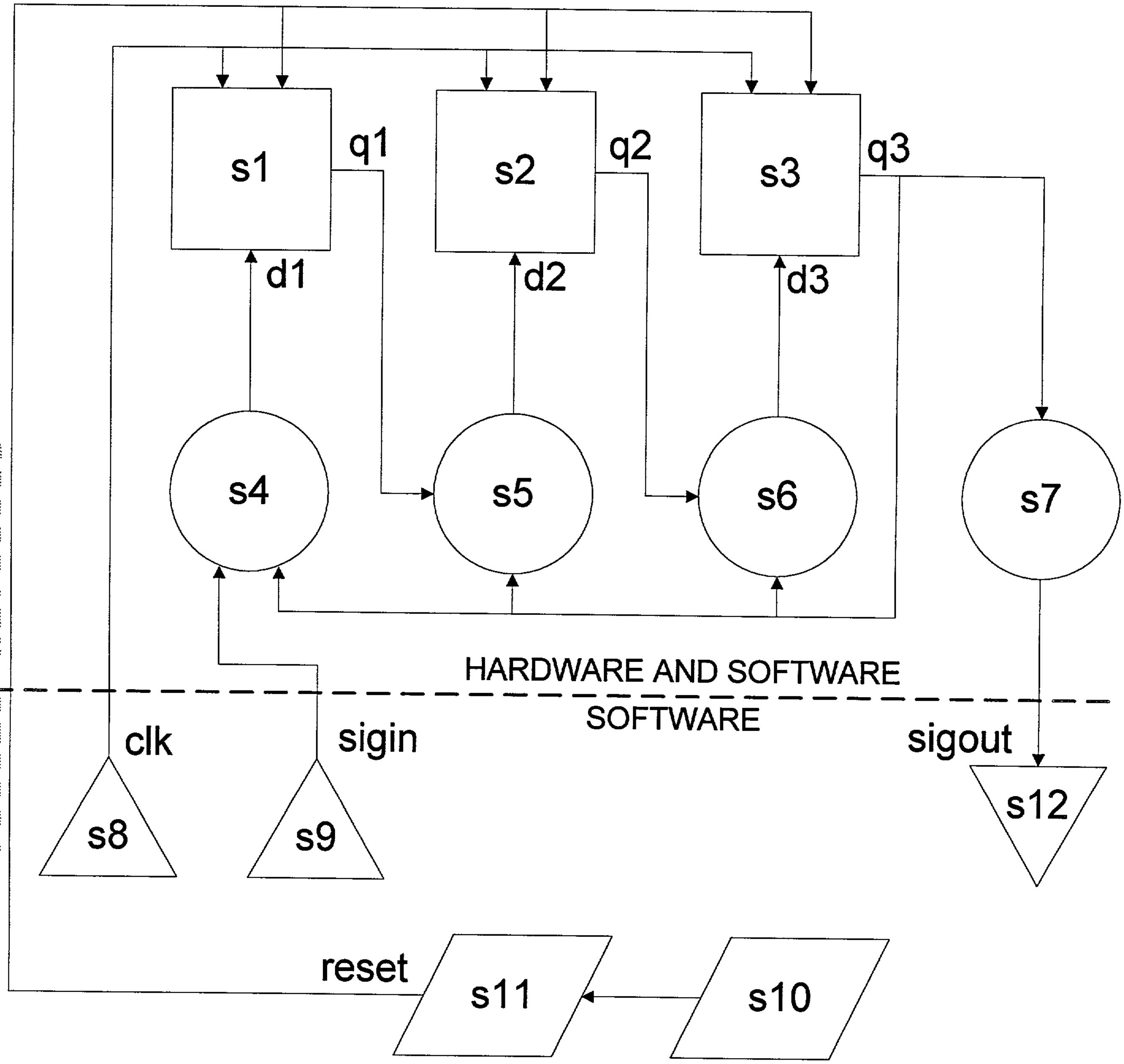


FIG. 29

# SOFTWARE/HARDWARE PARTITION RESULT

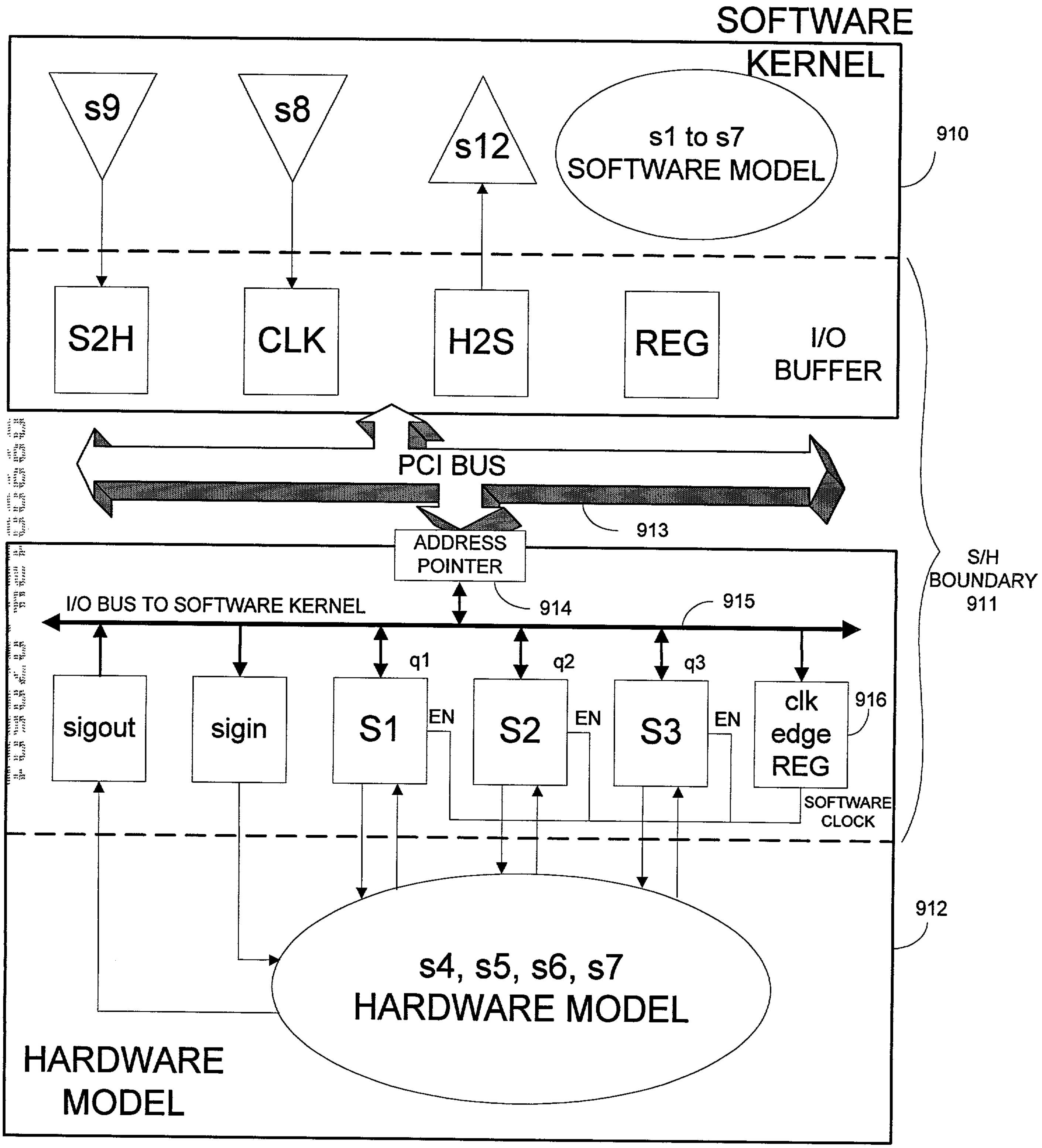


FIG. 30

# HARDWARE MODEL

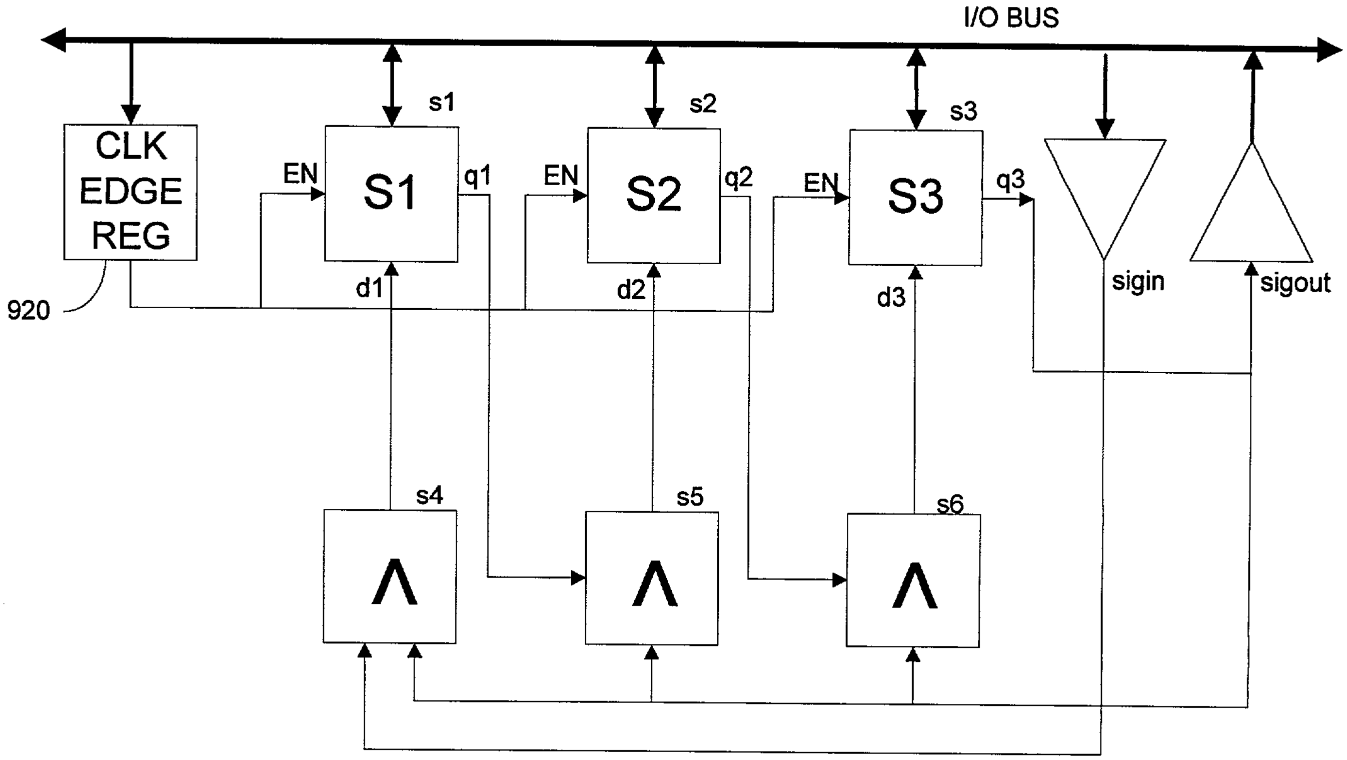
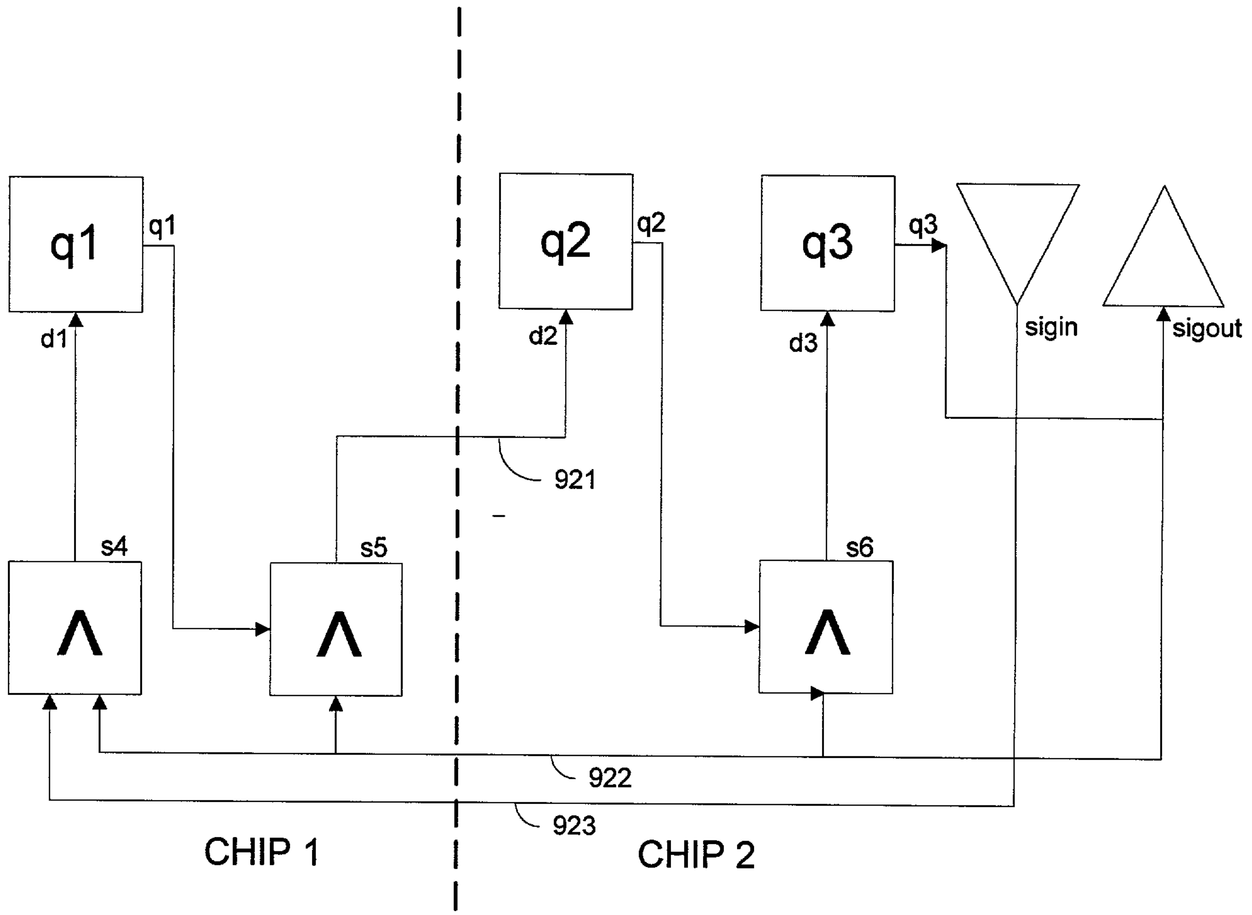


FIG. 31

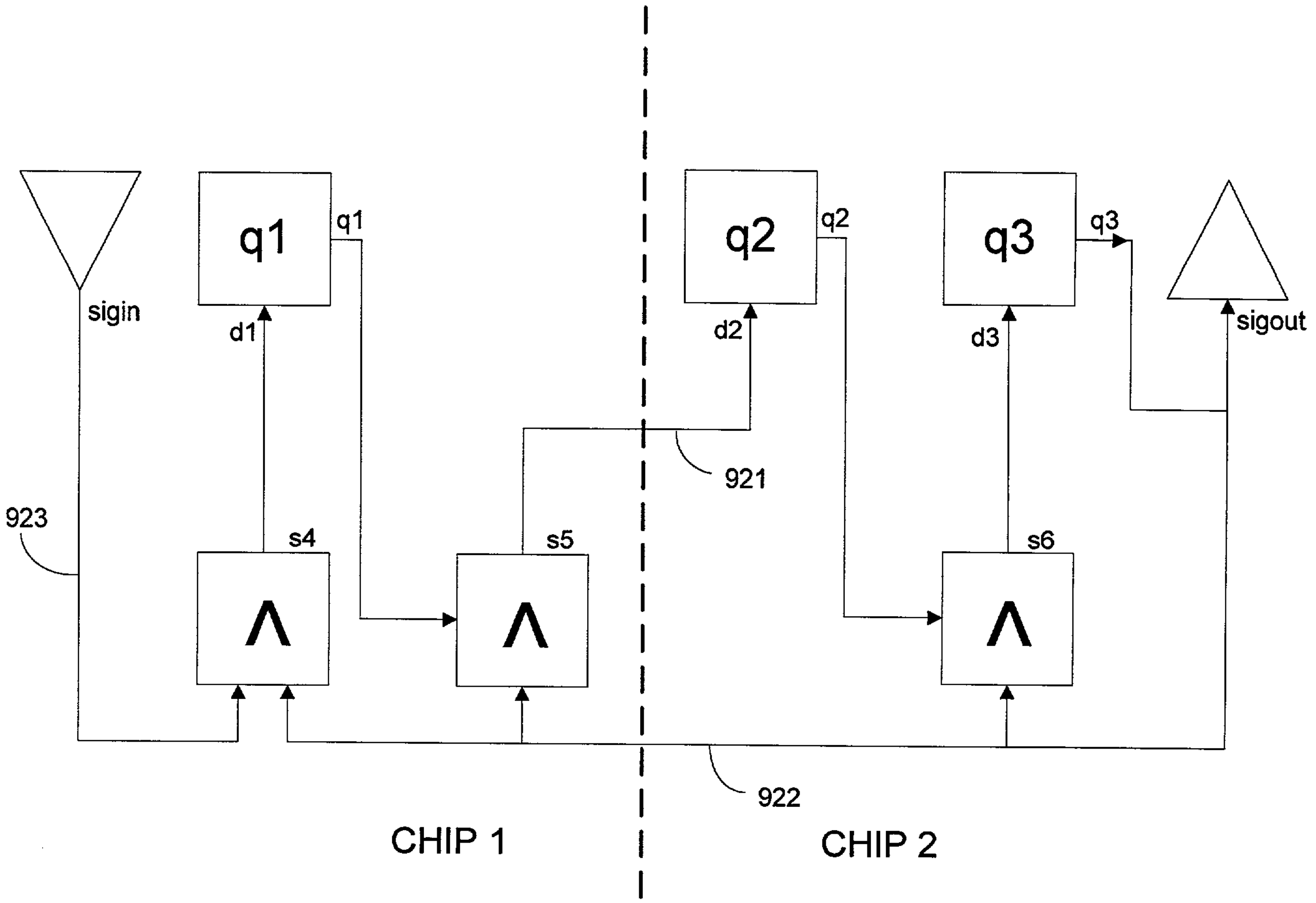
PARTITION RESULT #1



(IGNORE I/O AND CLOCK EDGE REGISTER)

FIG. 32

PARTITION RESULT #2



(IGNORE I/O AND CLOCK EDGE REGISTER)

FIG. 33

# LOGIC PATCHING

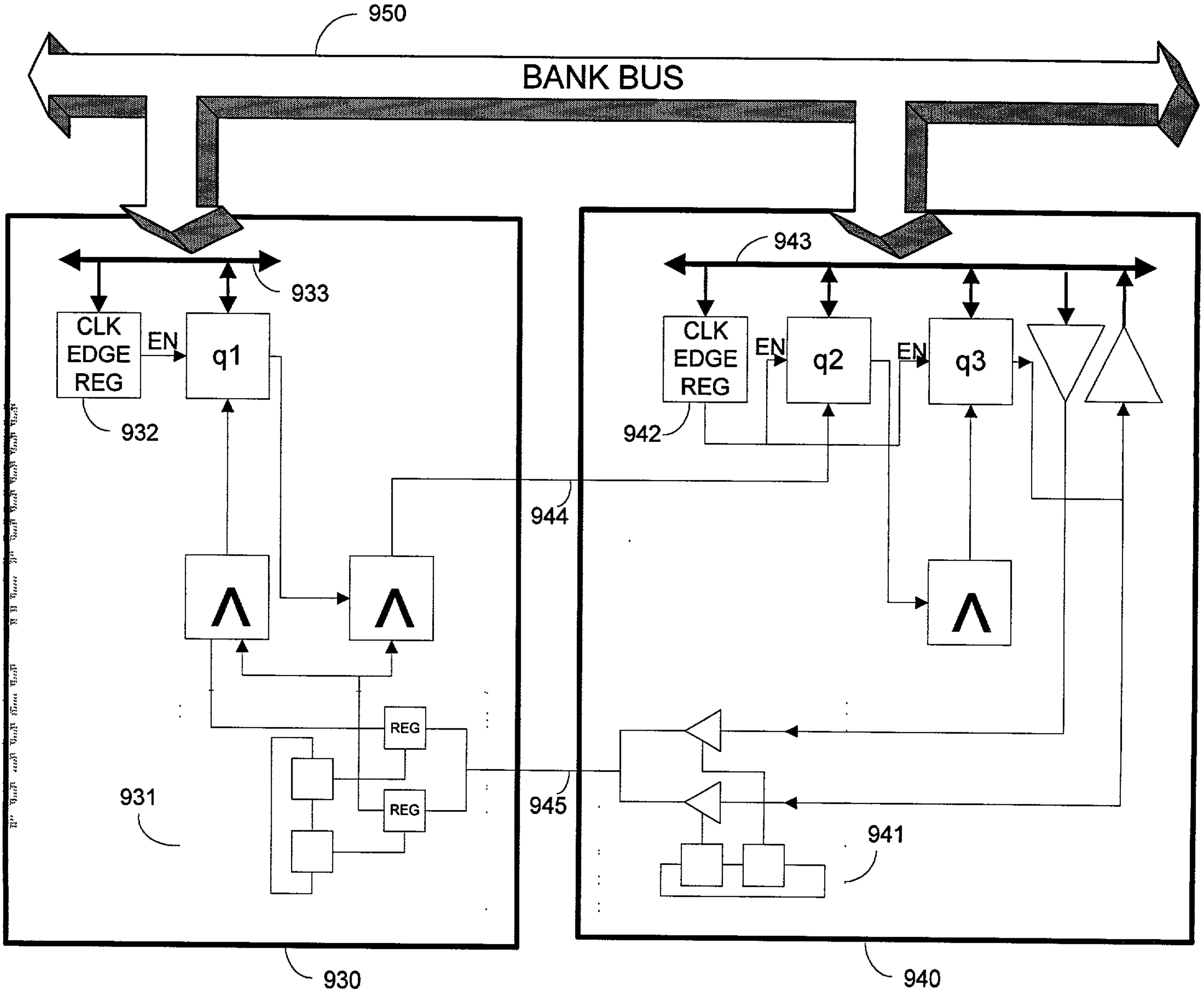
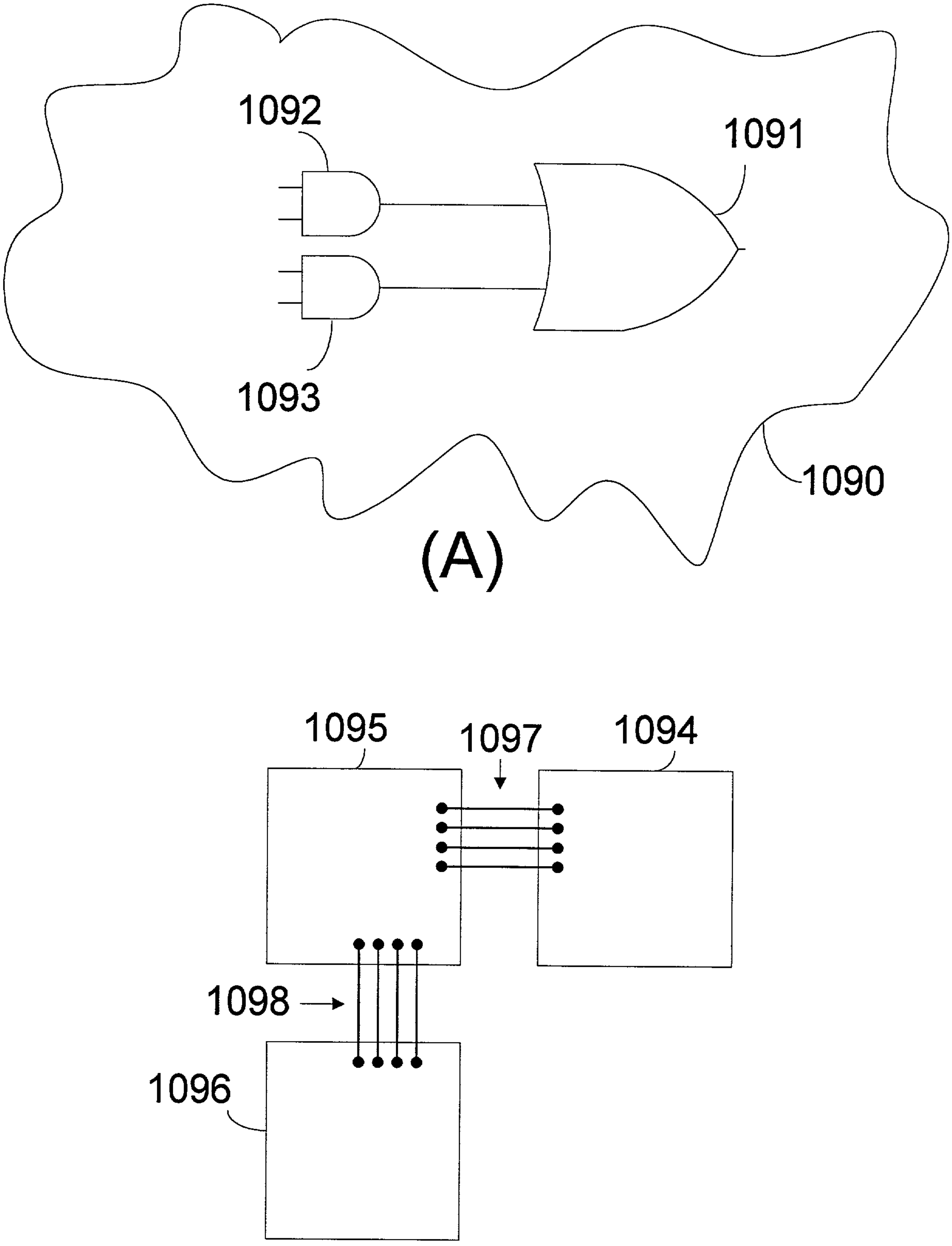


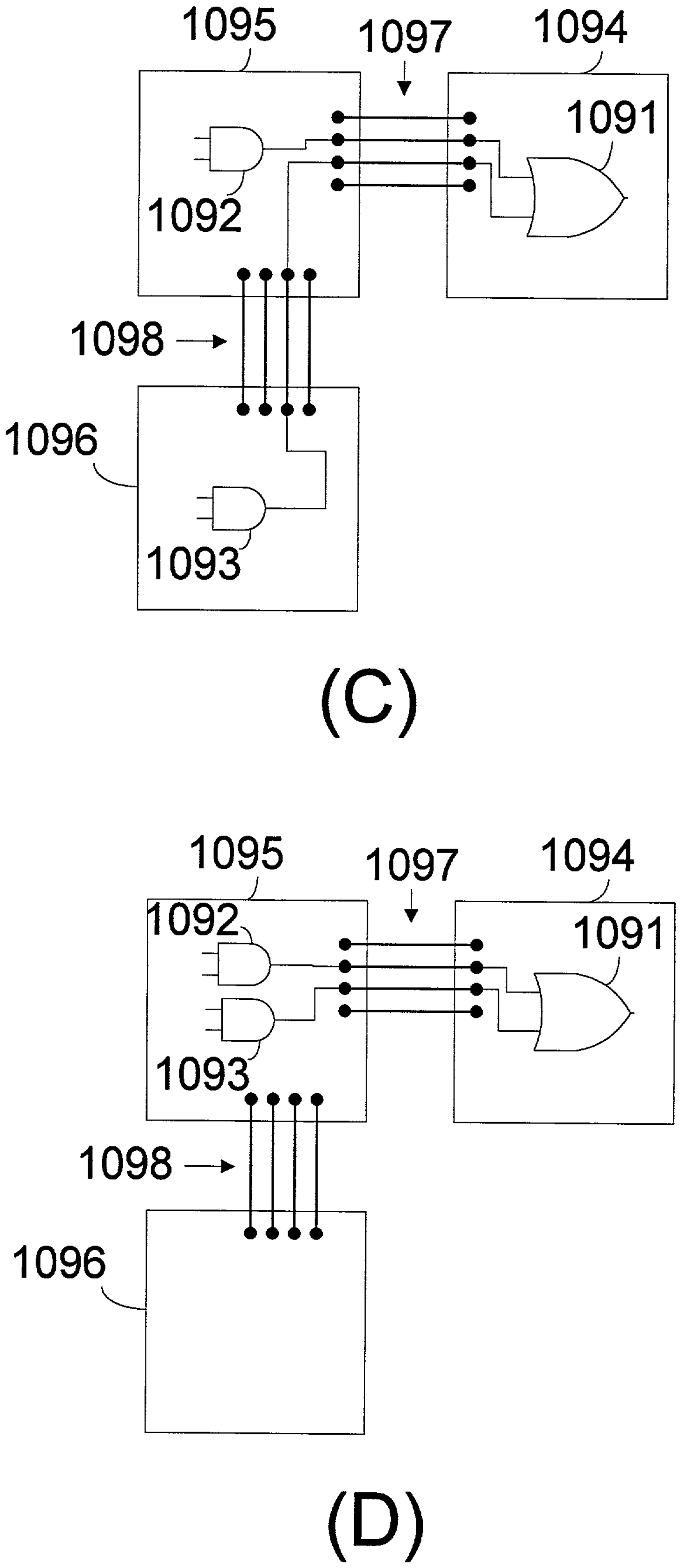
FIG. 34



(A)

(B)

FIG. 35


1095
1097
1094
1091
1092
1098
1096
1093
1095
1097
1094
1092
1091
1093
1098
1096

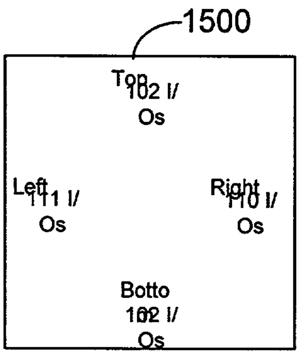
(C)

(D)

FIG. 35

# I/O PIN OVERVIEW OF FPGA LOGIC DEVICE

FPGA : 10K130V, 10K250V with 599-pin PGA package



*425 Interconnect I/O pins*

45 Dedicated I/O pins:

GCLK, BUS[31..0], F_RD,
FD_DATAXSFR, SHIFTIN, F_WR,
SPACE[2..0], SHIFTOUT, EVAL,
DEV_OE, DEV_REQ_N,
DEV_CLRN

FIG. 36

# FPGA INTERCONNECT BUSES

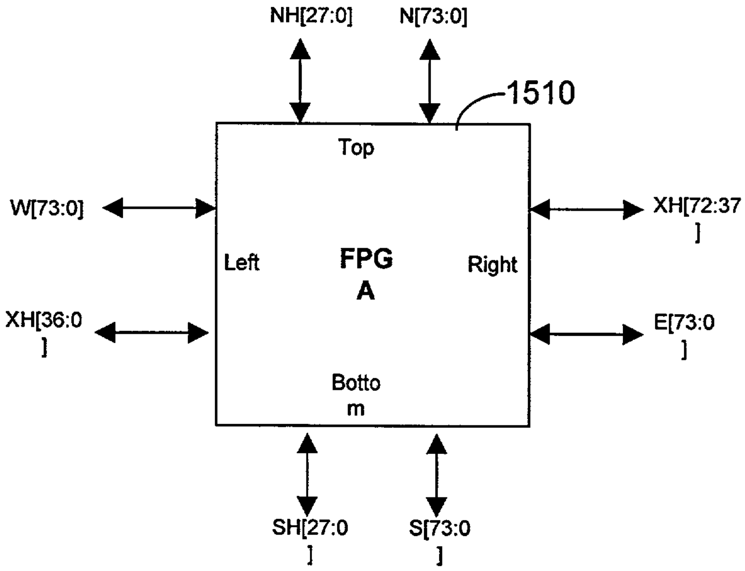


FIG. 37

BOARD CONNECTION - SIDE VIEW

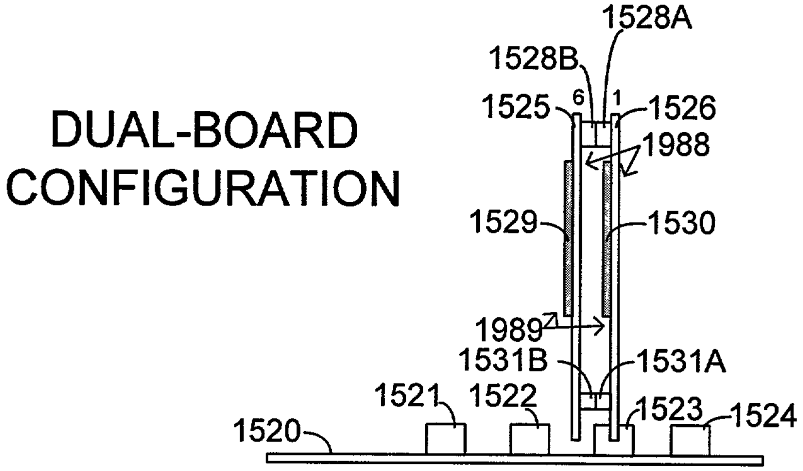


FIG. 38(A)

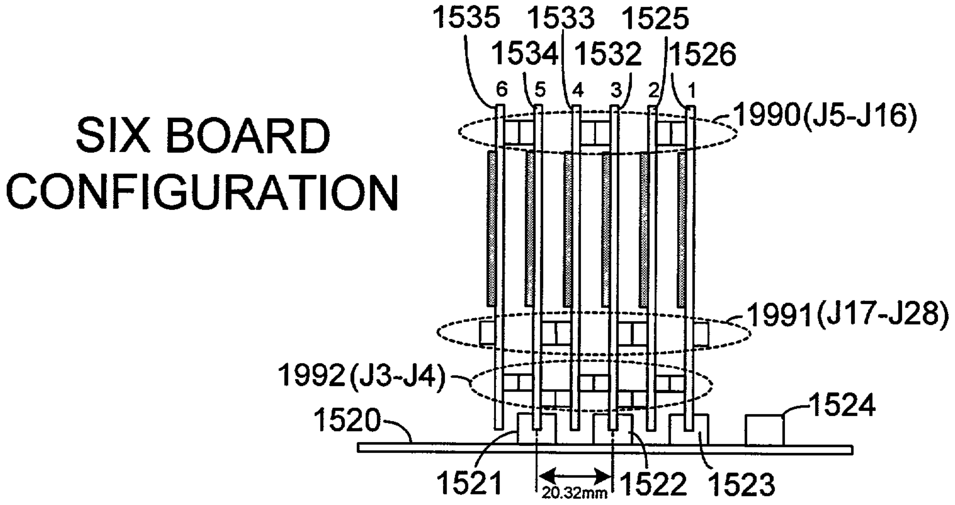


FIG. 38(B)

SIX-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

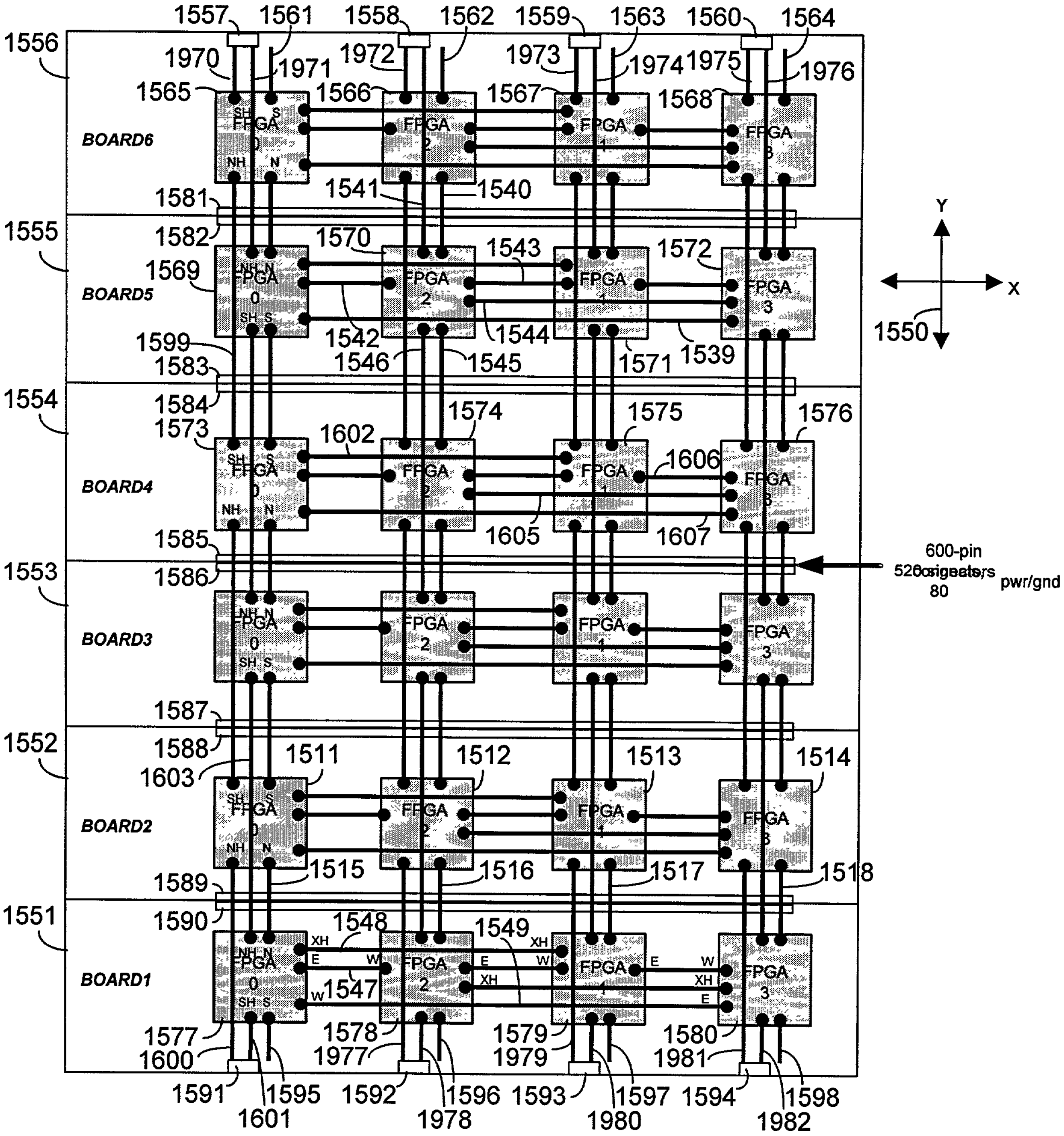


FIG. 39

FPGA ARRAY CONNECTION BETWEEN BOARDS

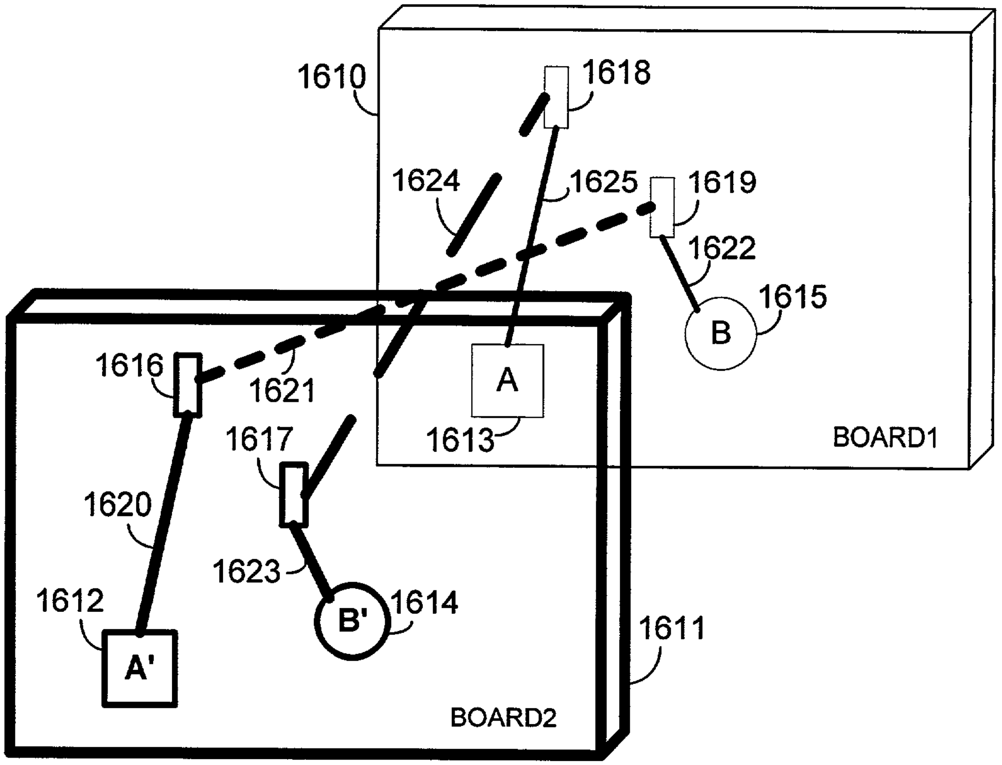


FIG. 40(A)

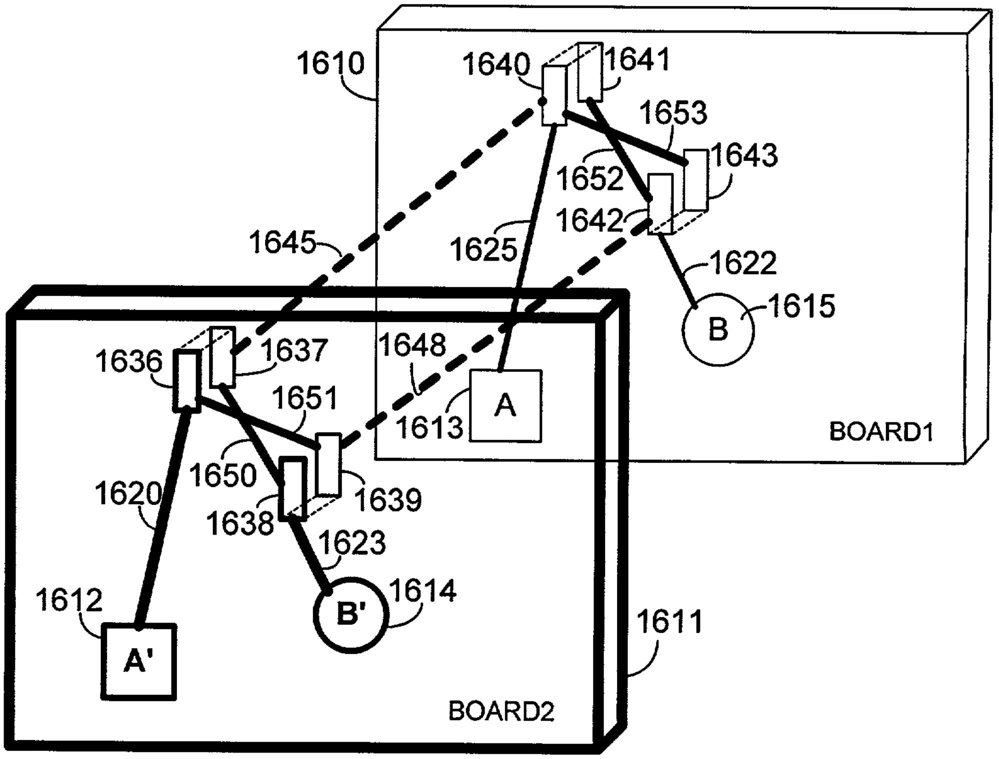


FIG. 40(B)

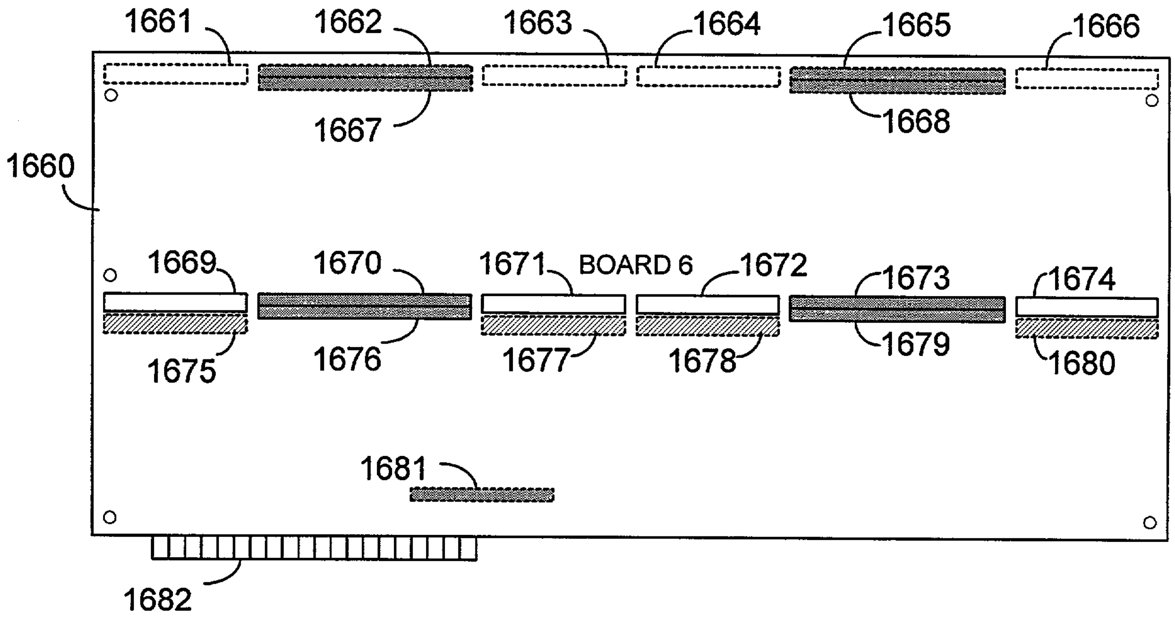


FIG. 41(A)

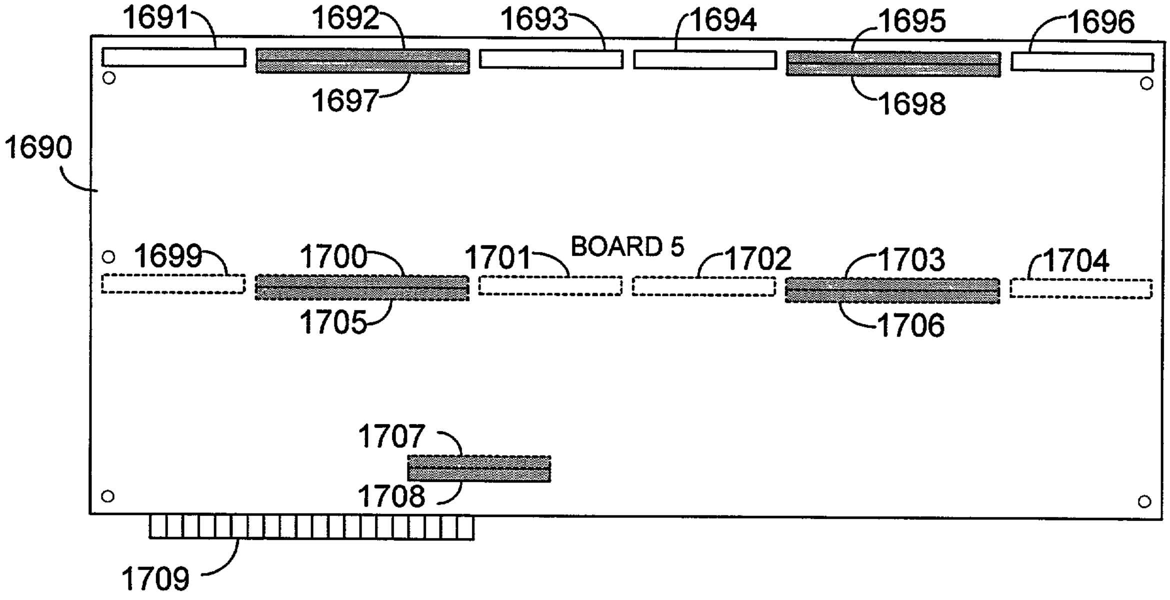


FIG. 41(B)

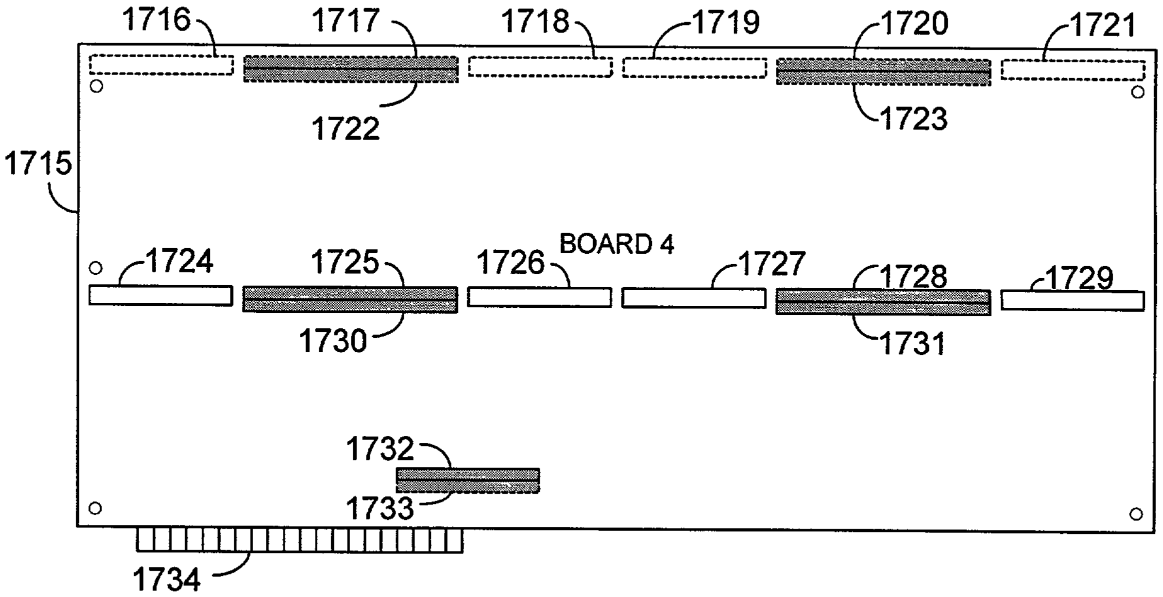


FIG. 41(C)

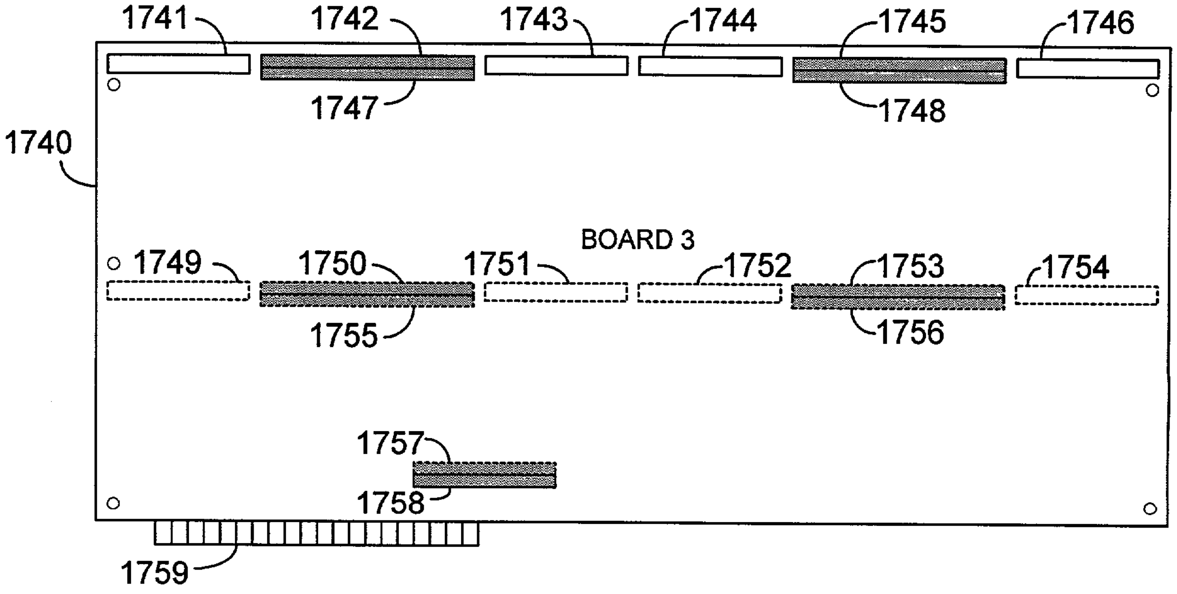


FIG. 41(D)

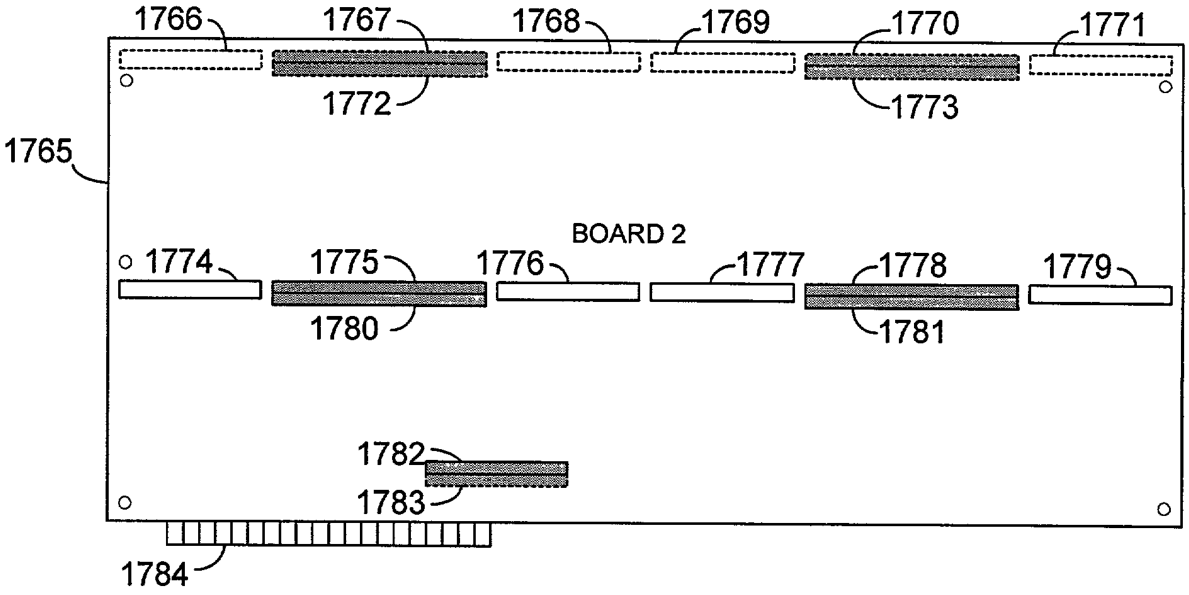


FIG. 41(E)

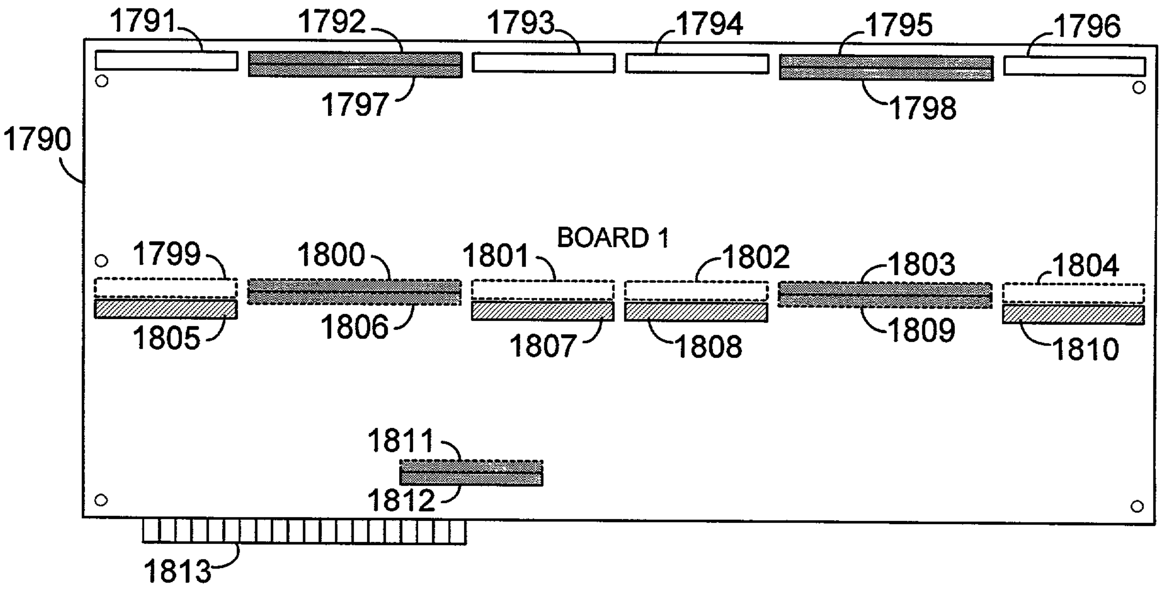


FIG. 41(F)

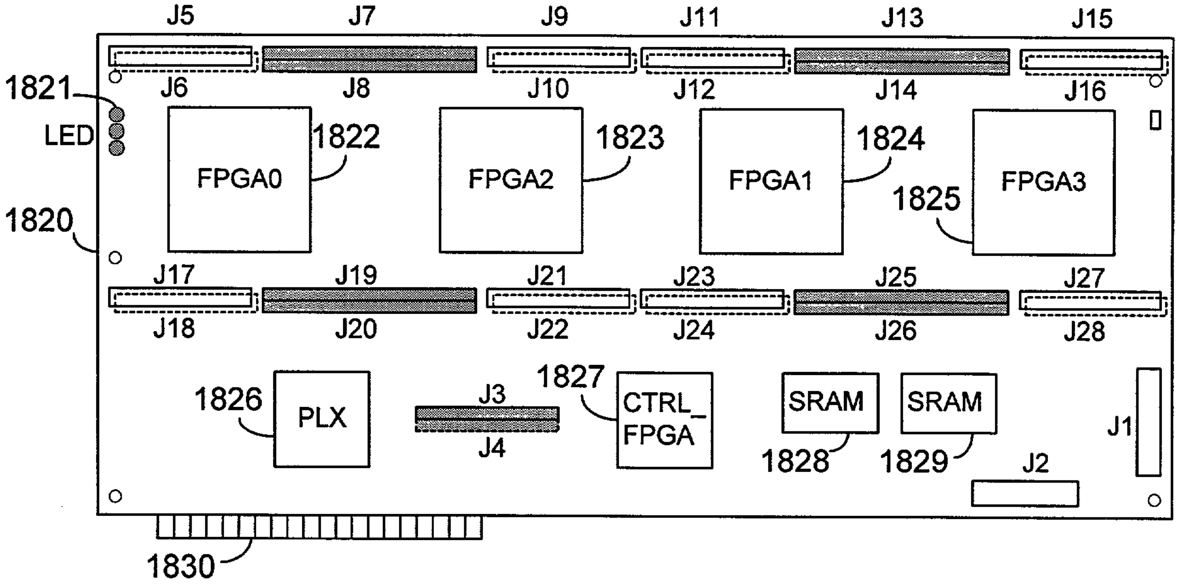

J5
J7
J9
J11
J13
J15
J6
J8
J10
J12
J14
J16
1821
LED
1822
1823
1824
FPGA0
FPGA2
FPGA1
FPGA3
1825
1820
J17
J19
J21
J23
J25
J27
J18
J20
J22
J24
J26
J28
1827
1826
PLX
J3
J4
CTRL_
FPGA
SRAM
SRAM
J1
1828
1829
J2
1830

FIG. 42

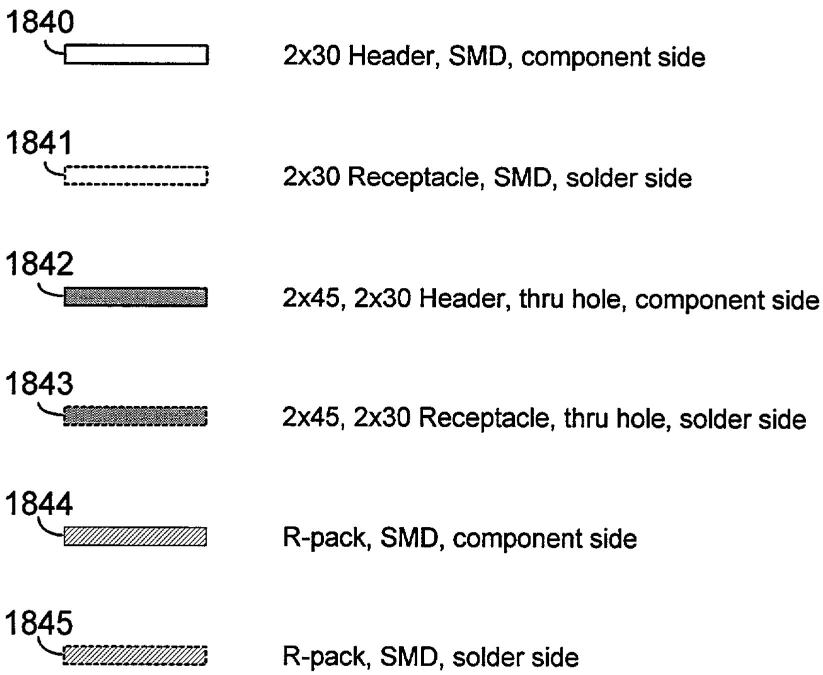

1840
2x30 Header, SMD, component side
1841
2x30 Receptacle, SMD, solder side
1842
2x45, 2x30 Header, thru hole, component side
1843
2x45, 2x30 Receptacle, thru hole, solder side
1844
R-pack, SMD, component side
1845
R-pack, SMD, solder side

FIG. 43

TWO-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

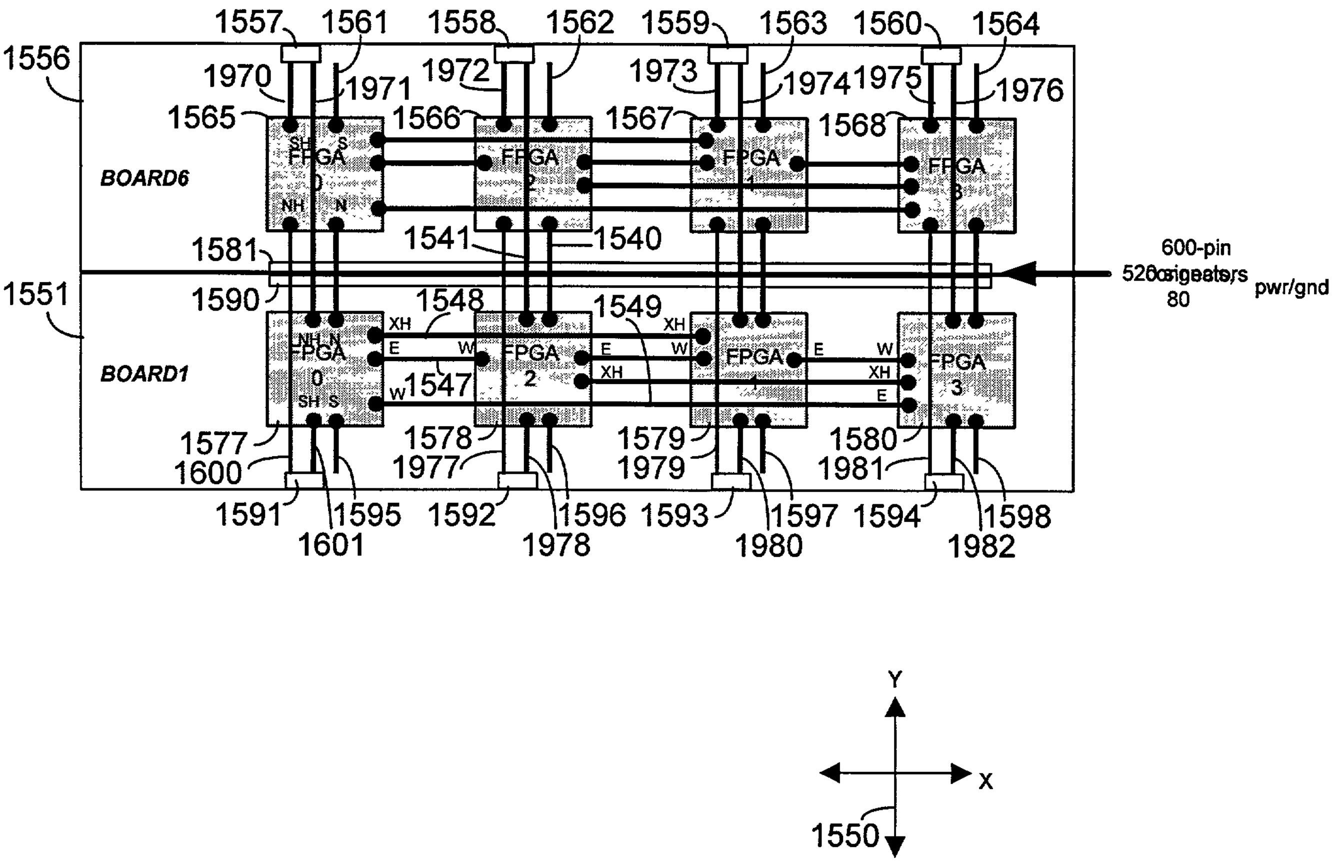


FIG. 44

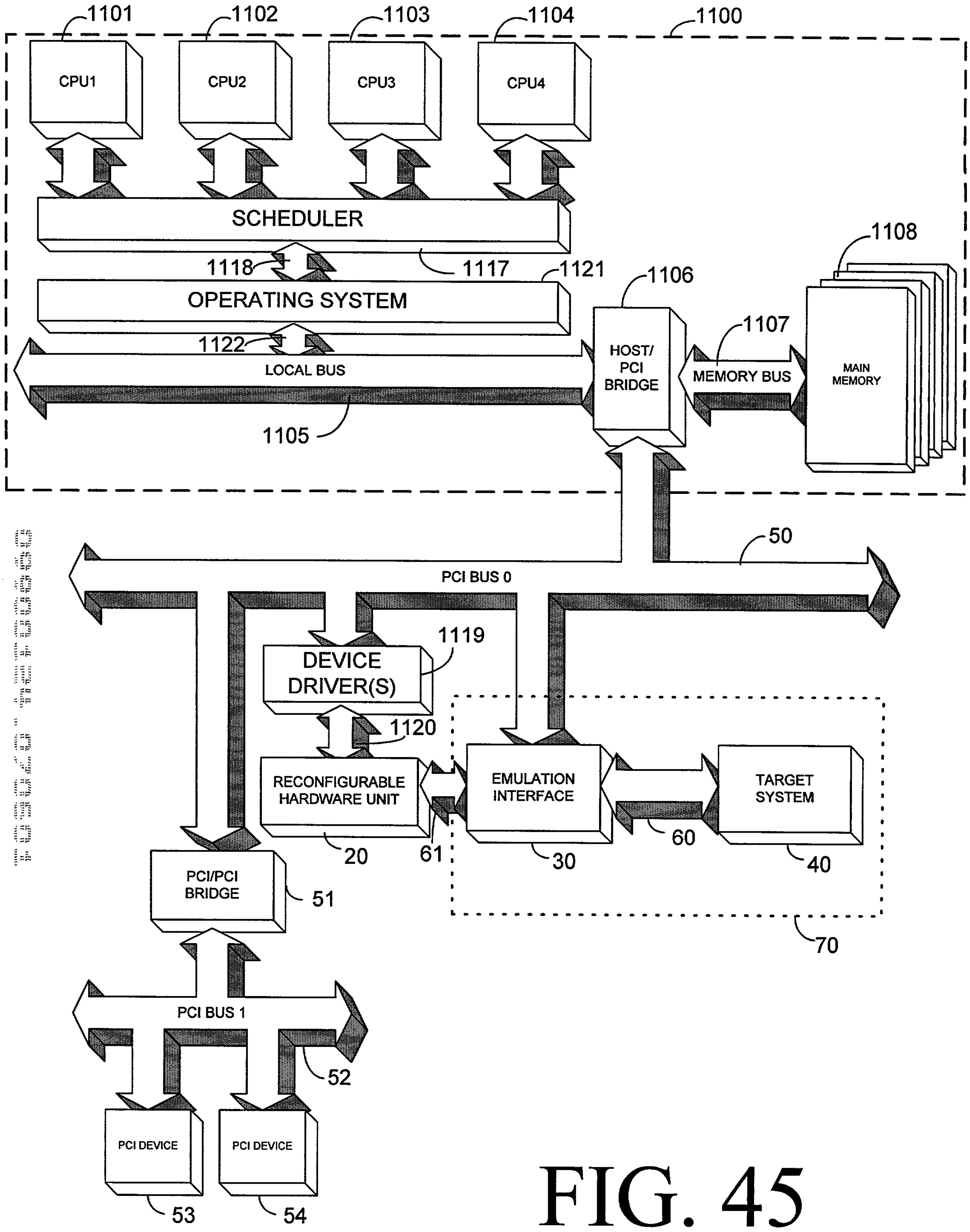

1101
1102
1103
1104
1100
CPU1
CPU2
CPU3
CPU4
SCHEDULER
1118
1117
1121
OPERATING SYSTEM
1106
1108
1122
1107
LOCAL BUS
HOST/
PCI
BRIDGE
MEMORY BUS
MAIN
MEMORY
1105
50
PCI BUS 0
1119
DEVICE
DRIVER(S)
1120
RECONFIGURABLE
HARDWARE UNIT
EMULATION
INTERFACE
TARGET
SYSTEM
60
20
61
30
40
PCI/PCI
BRIDGE
51
70
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG. 45

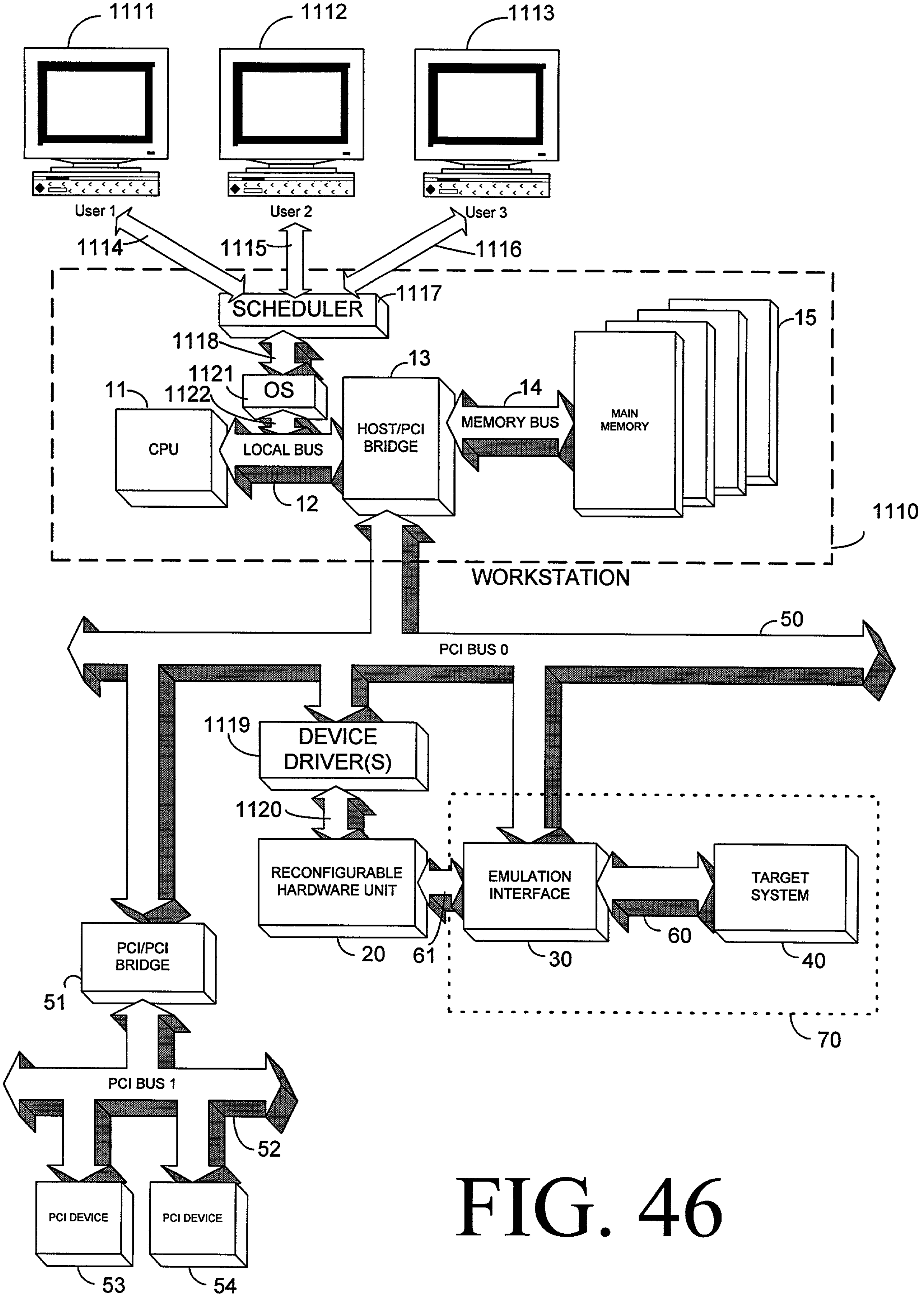

1111
1112
1113
User 1
User 2
User 3
1114
1115
1116
SCHEDULER
1117
1118
1121
OS
1122
13
14
15
11
CPU
LOCAL BUS
HOST/PCI BRIDGE
MEMORY BUS
MAIN MEMORY
12
1110
WORKSTATION
50
PCI BUS 0
1119
DEVICE DRIVER(S)
1120
RECONFIGURABLE HARDWARE UNIT
EMULATION INTERFACE
TARGET SYSTEM
20
61
30
60
40
70
PCI/PCI BRIDGE
51
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG. 46

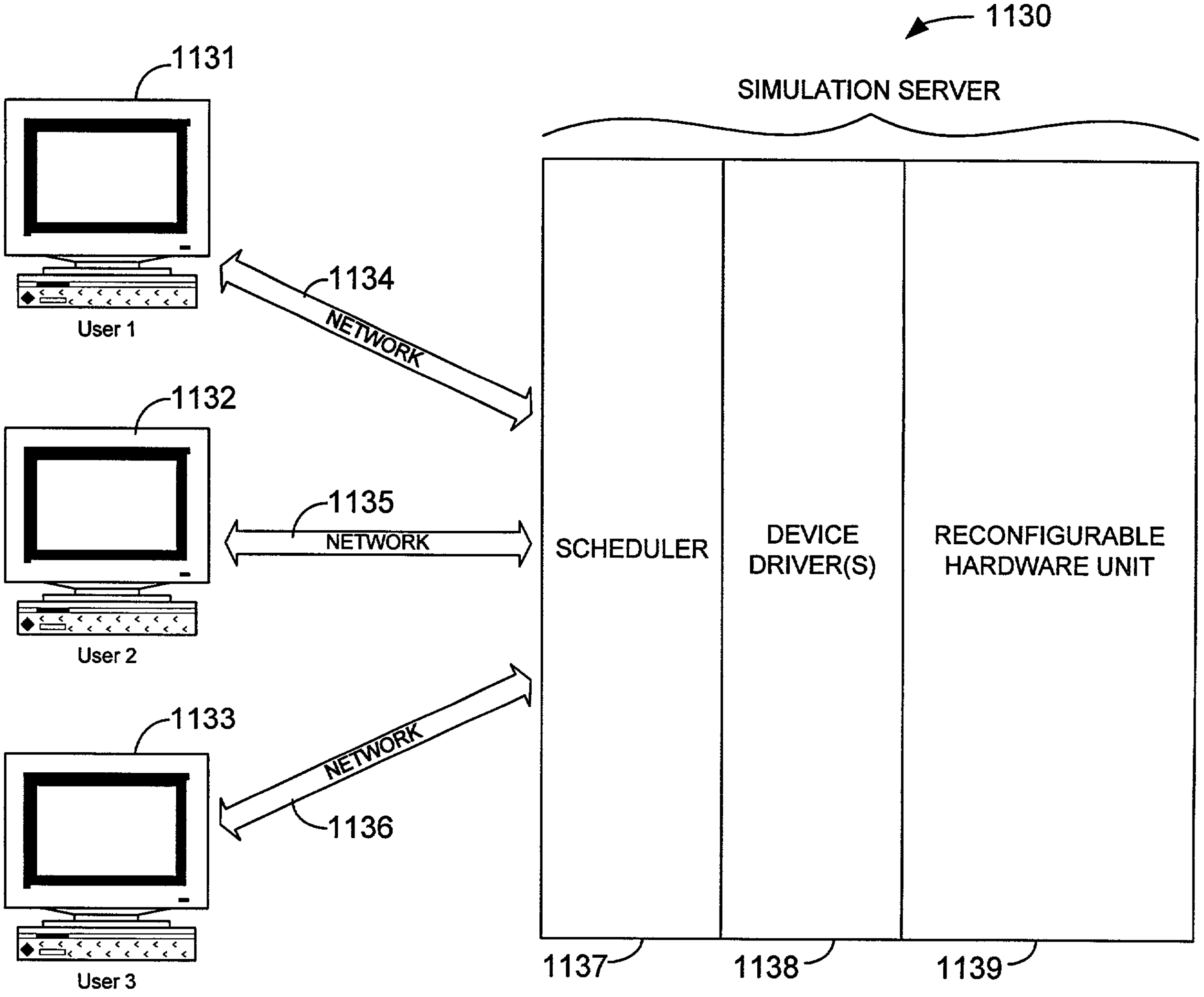

1130
SIMULATION SERVER
1131
User 1
1134
NETWORK
1132
User 2
1135
NETWORK
1133
User 3
NETWORK
1136
SCHEDULER
DEVICE DRIVER(S)
RECONFIGURABLE HARDWARE UNIT
1137
1138
1139

FIG. 47

# SIMULATION SERVER ARCHITECTURE

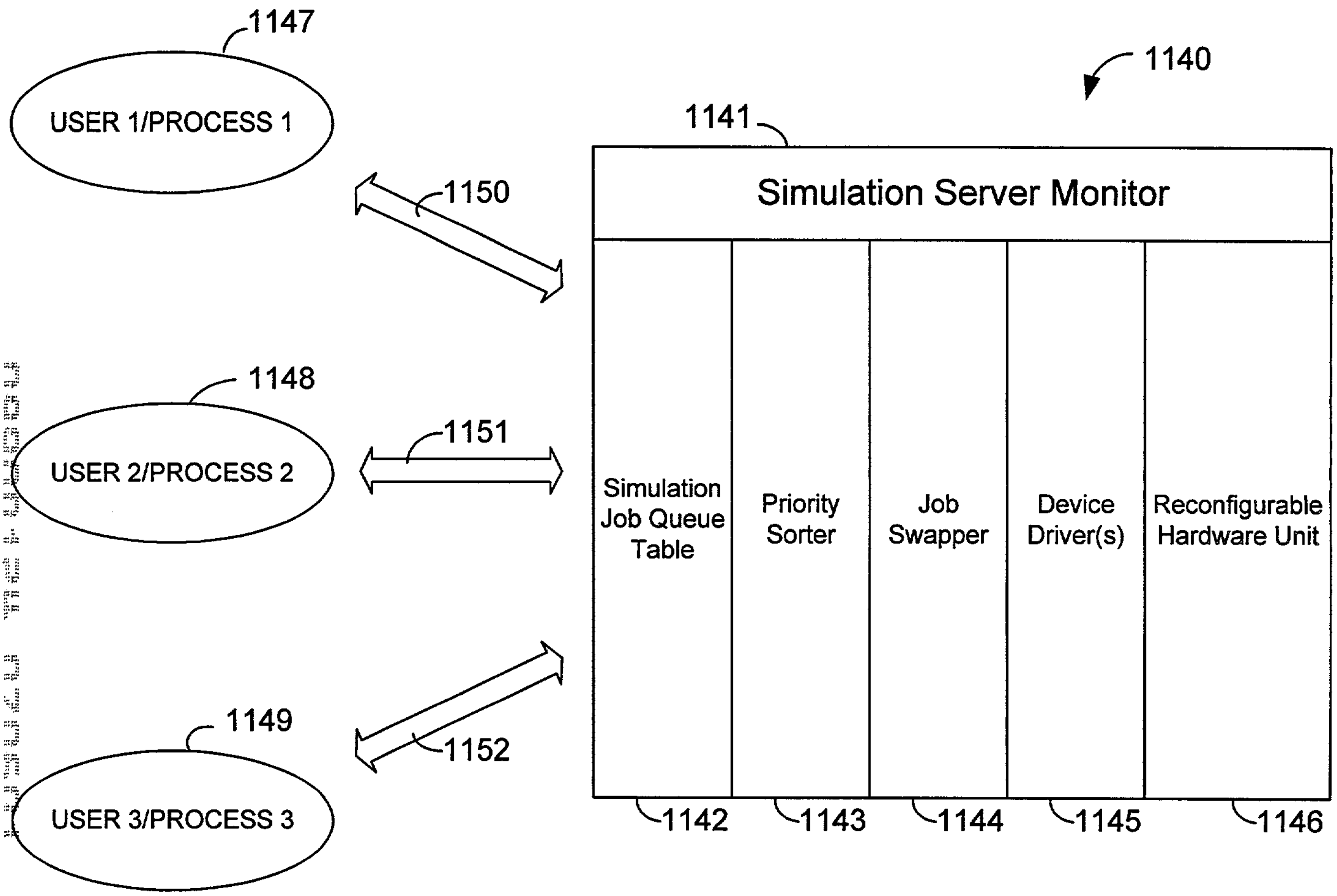


FIG. 48

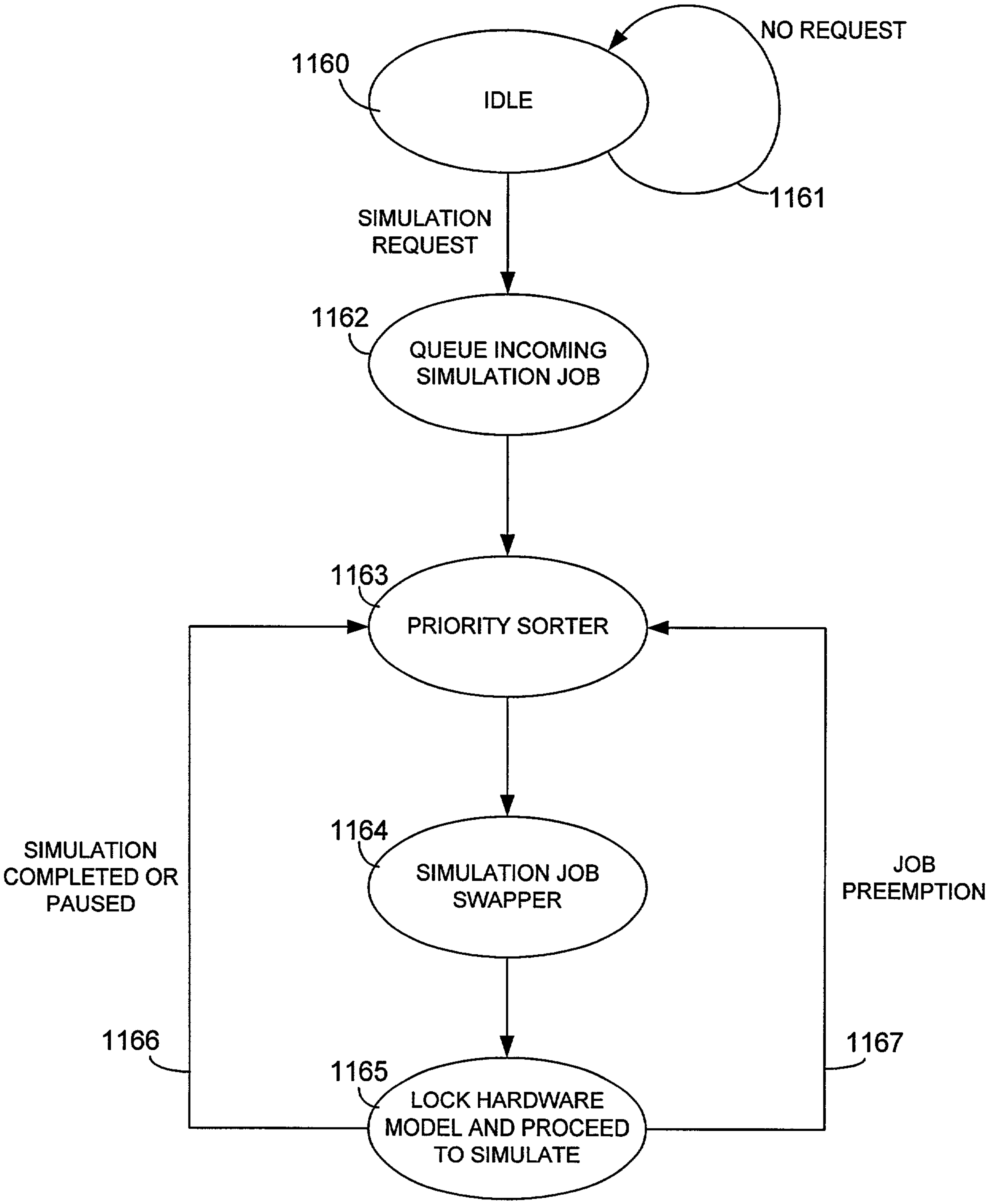

NO REQUEST
1160
IDLE
1161
SIMULATION REQUEST
1162
QUEUE INCOMING SIMULATION JOB
1163
PRIORITY SORTER
1164
SIMULATION JOB SWAPPER
SIMULATION COMPLETED OR PAUSED
JOB PREEMPTION
1166
1165
LOCK HARDWARE MODEL AND PROCEED TO SIMULATE
1167

FIG. 49

# JOB SWAPPER

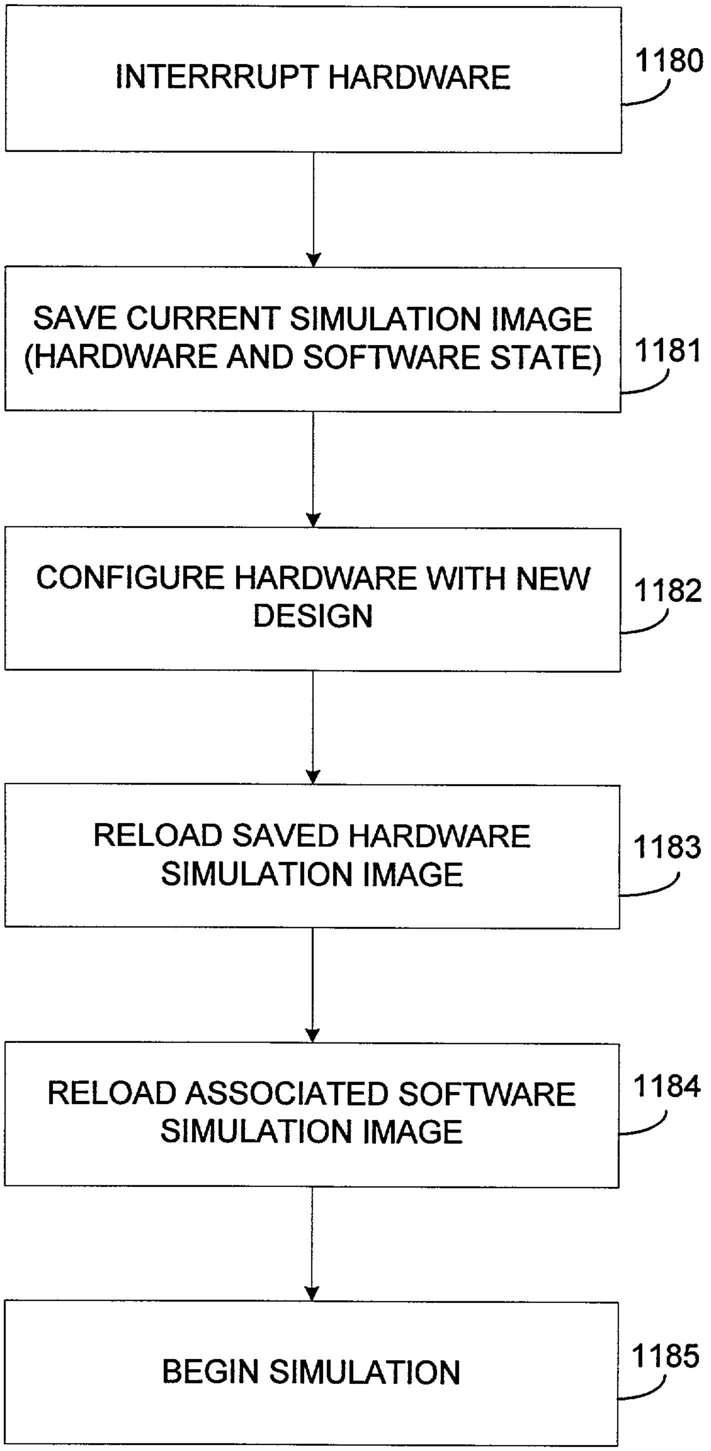

INTERRRUPT HARDWARE
1180
SAVE CURRENT SIMULATION IMAGE (HARDWARE AND SOFTWARE STATE)
1181
CONFIGURE HARDWARE WITH NEW DESIGN
1182
RELOAD SAVED HARDWARE SIMULATION IMAGE
1183
RELOAD ASSOCIATED SOFTWARE SIMULATION IMAGE
1184
BEGIN SIMULATION
1185

FIG. 50

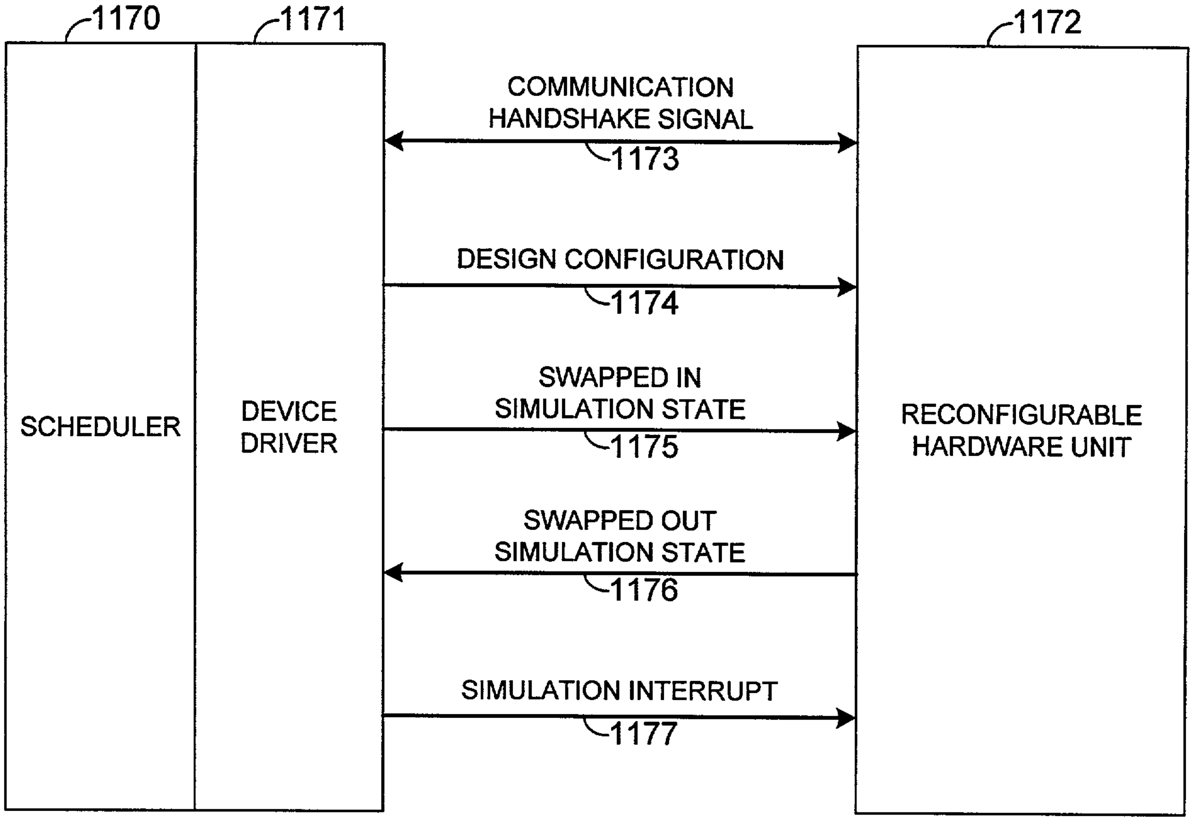

1170
1171
1172
SCHEDULER
DEVICE DRIVER
RECONFIGURABLE HARDWARE UNIT
COMMUNICATION HANDSHAKE SIGNAL
1173
DESIGN CONFIGURATION
1174
SWAPPED IN SIMULATION STATE
1175
SWAPPED OUT SIMULATION STATE
1176
SIMULATION INTERRUPT
1177

FIG. 51

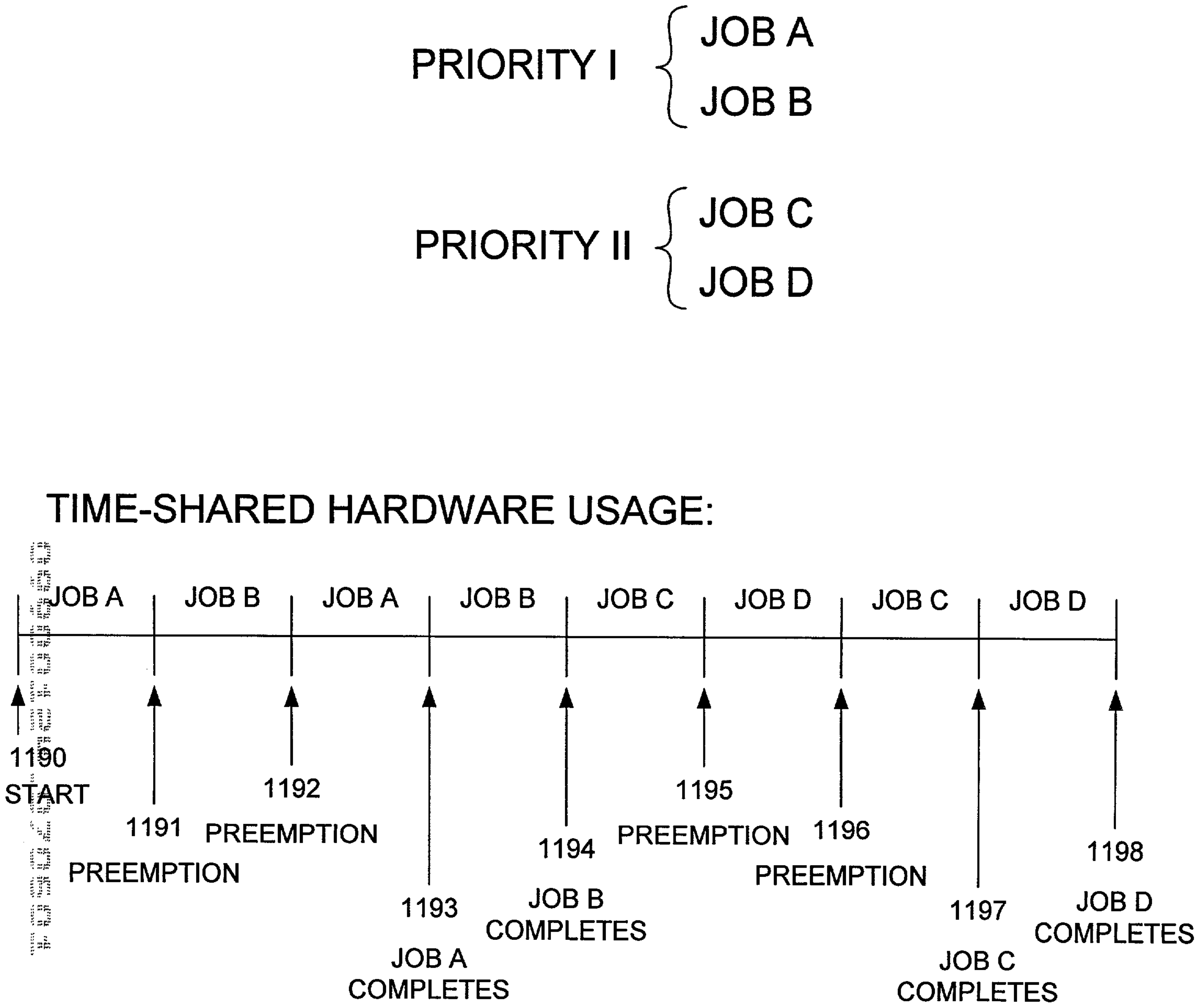

PRIORITY I
JOB A
JOB B
PRIORITY II
JOB C
JOB D
TIME-SHARED HARDWARE USAGE:
JOB A
JOB B
JOB A
JOB B
JOB C
JOB D
JOB C
JOB D
1190
START
1191
PREEMPTION
1192
PREEMPTION
1193
JOB A
COMPLETES
1194
JOB B
COMPLETES
1195
PREEMPTION
1196
PREEMPTION
1197
JOB C
COMPLETES
1198
JOB D
COMPLETES

FIG. 52

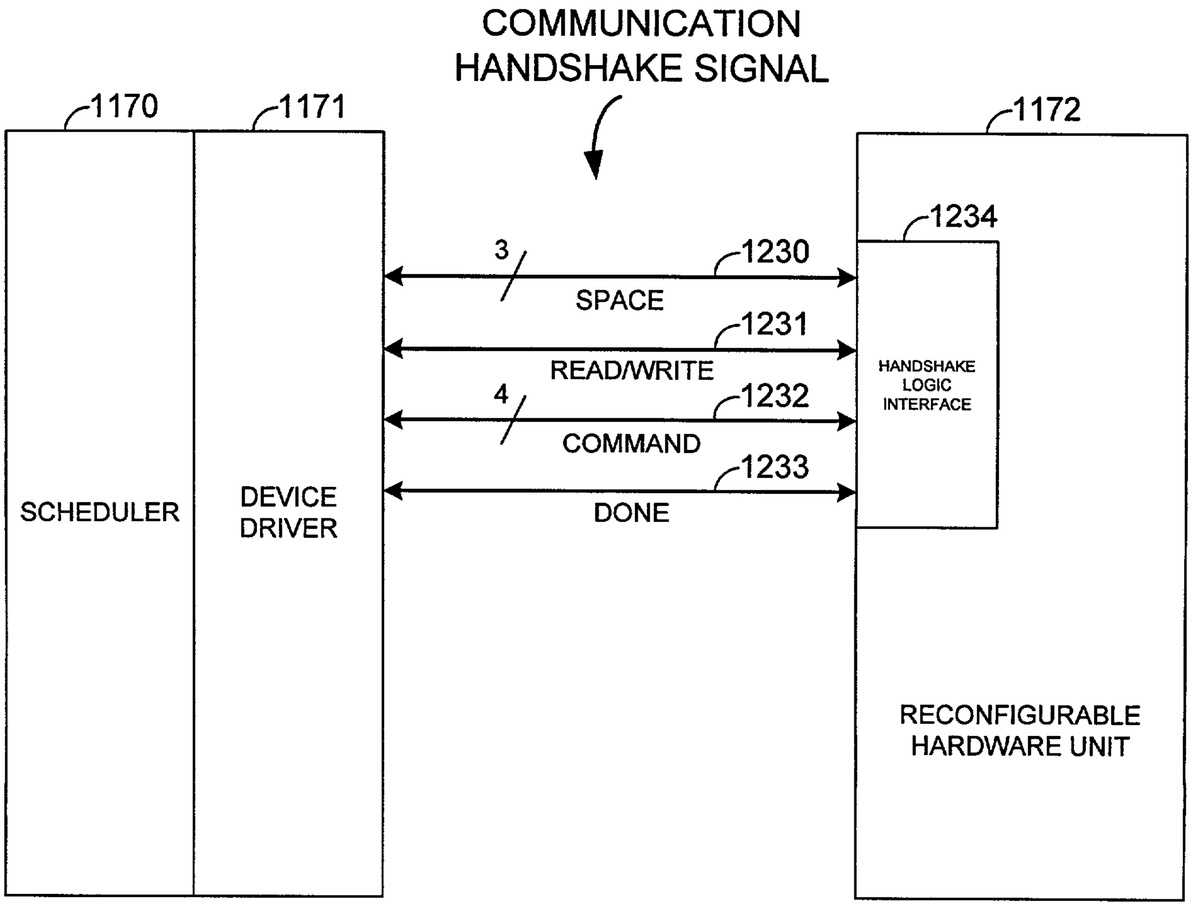

COMMUNICATION
HANDSHAKE SIGNAL
1170
1171
1172
1234
3
1230
SPACE
1231
READ/WRITE
4
1232
COMMAND
1233
DONE
HANDSHAKE
LOGIC
INTERFACE
SCHEDULER
DEVICE
DRIVER
RECONFIGURABLE
HARDWARE UNIT

FIG. 53

COMMUNICATION HANDSHAKE PROTOCOL

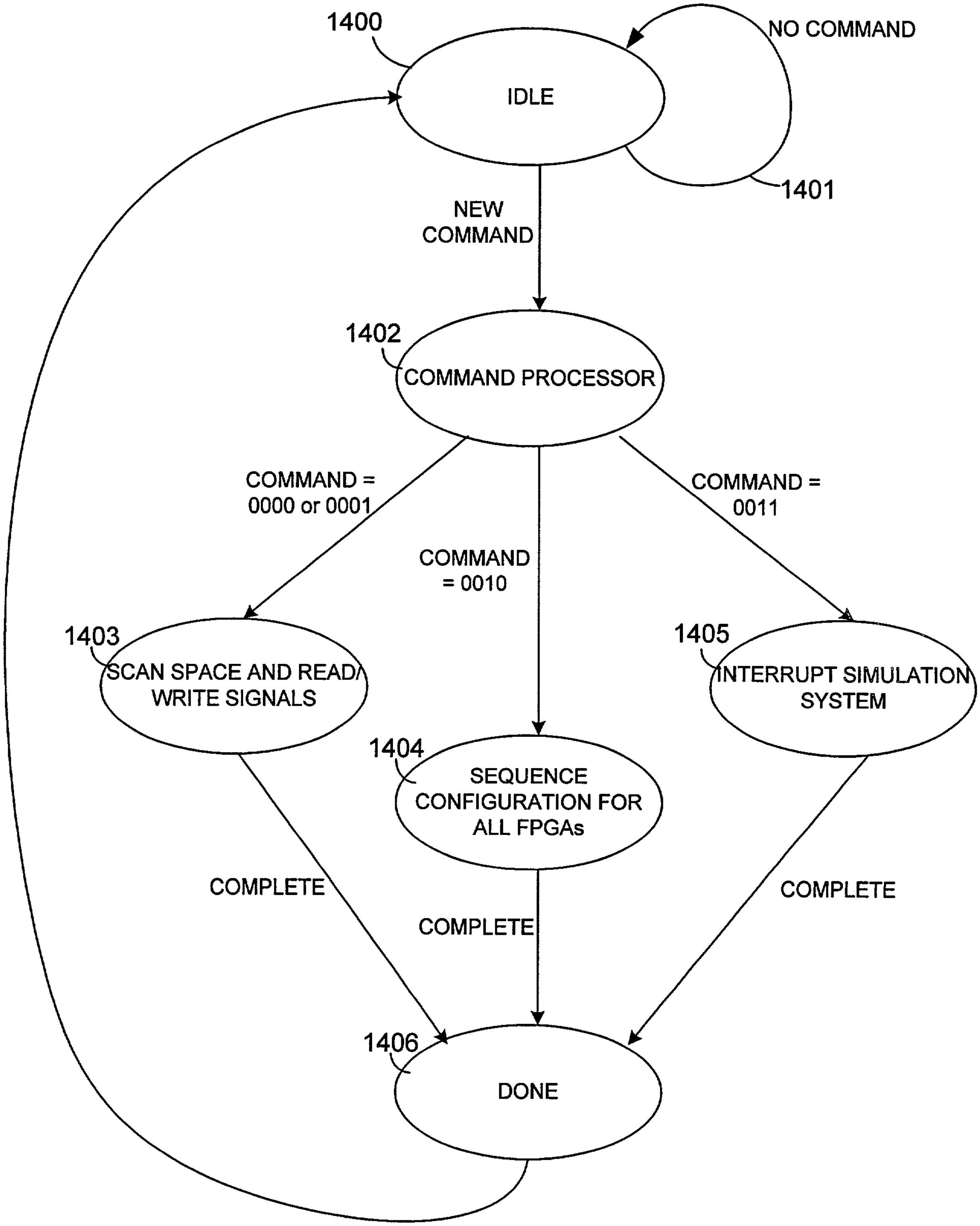


FIG. 54

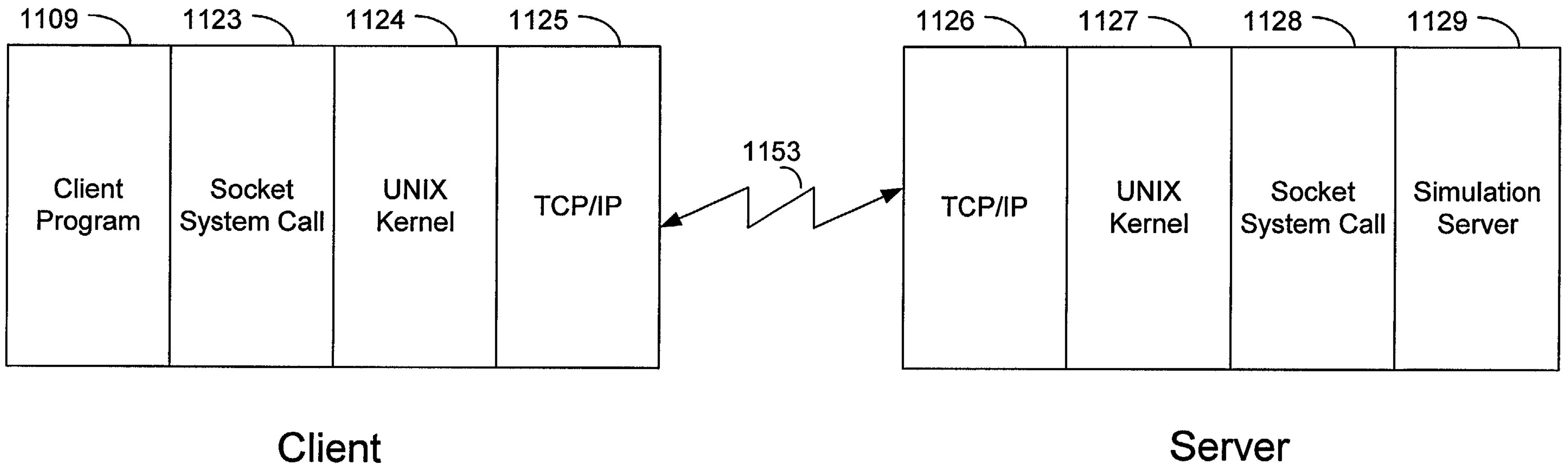

1109
1123
1124
1125
1153
1126
1127
1128
1129
Client Program
Socket System Call
UNIX Kernel
TCP/IP
TCP/IP
UNIX Kernel
Socket System Call
Simulation Server
Client
Server

FIG. 55

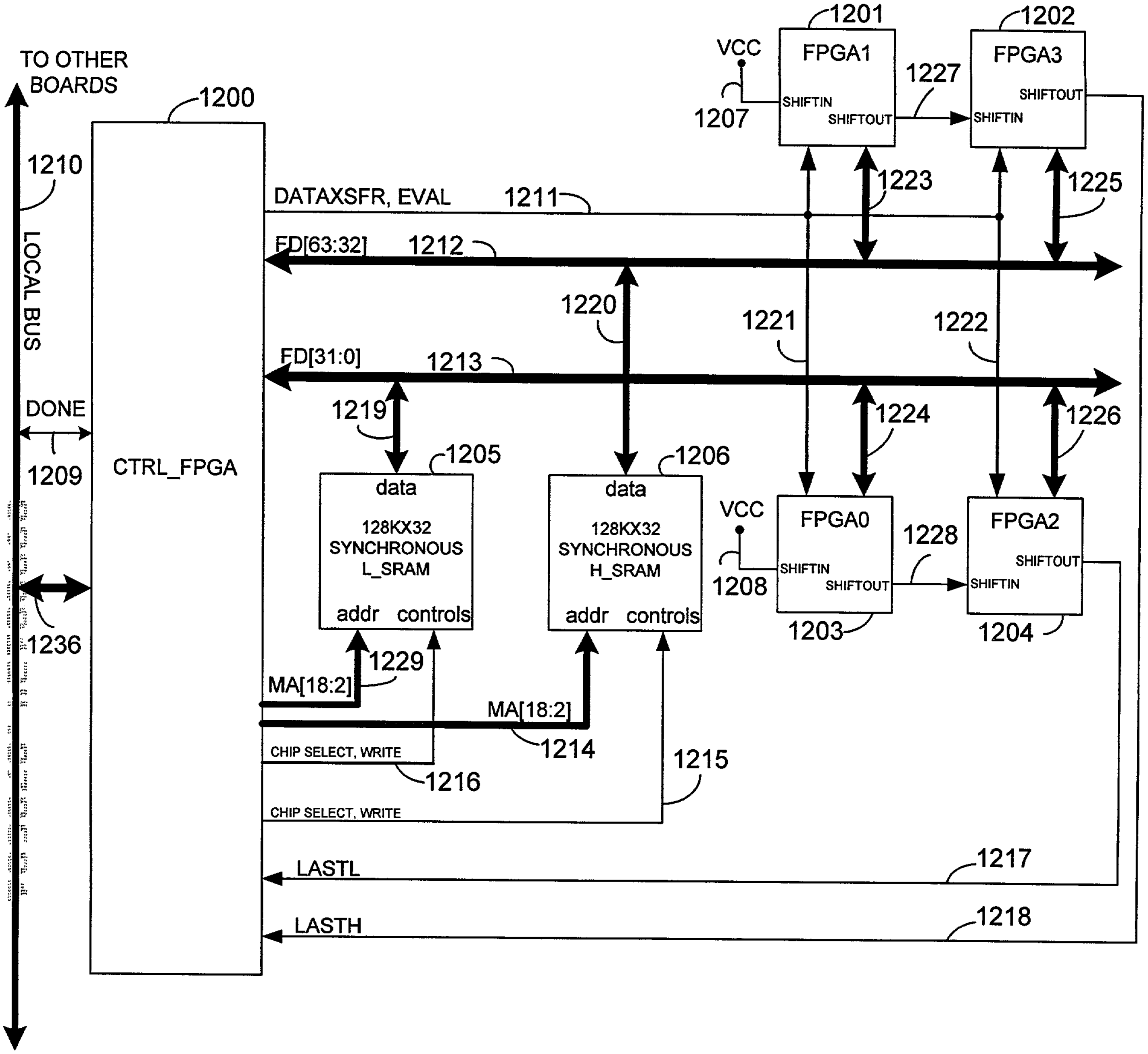

TO OTHER BOARDS
1210
LOCAL BUS
DONE
1209
1236
1200
CTRL_FPGA
DATAXSFR, EVAL
1211
FD[63:32]
1212
1220
FD[31:0]
1213
1219
1205
data
128KX32
SYNCHRONOUS
L_SRAM
addr
controls
1206
data
128KX32
SYNCHRONOUS
H_SRAM
addr
controls
MA[18:2]
1229
MA[18:2]
1214
CHIP SELECT, WRITE
1216
CHIP SELECT, WRITE
1215
LASTL
1217
LASTH
1218
1201
FPGA1
VCC
1207
SHIFTIN
SHIFTOUT
1227
1202
FPGA3
SHIFTOUT
SHIFTIN
1223
1225
1221
1222
1224
1226
VCC
1208
FPGA0
SHIFTIN
SHIFTOUT
1228
FPGA2
SHIFTOUT
SHIFTIN
1203
1204

FIG. 56

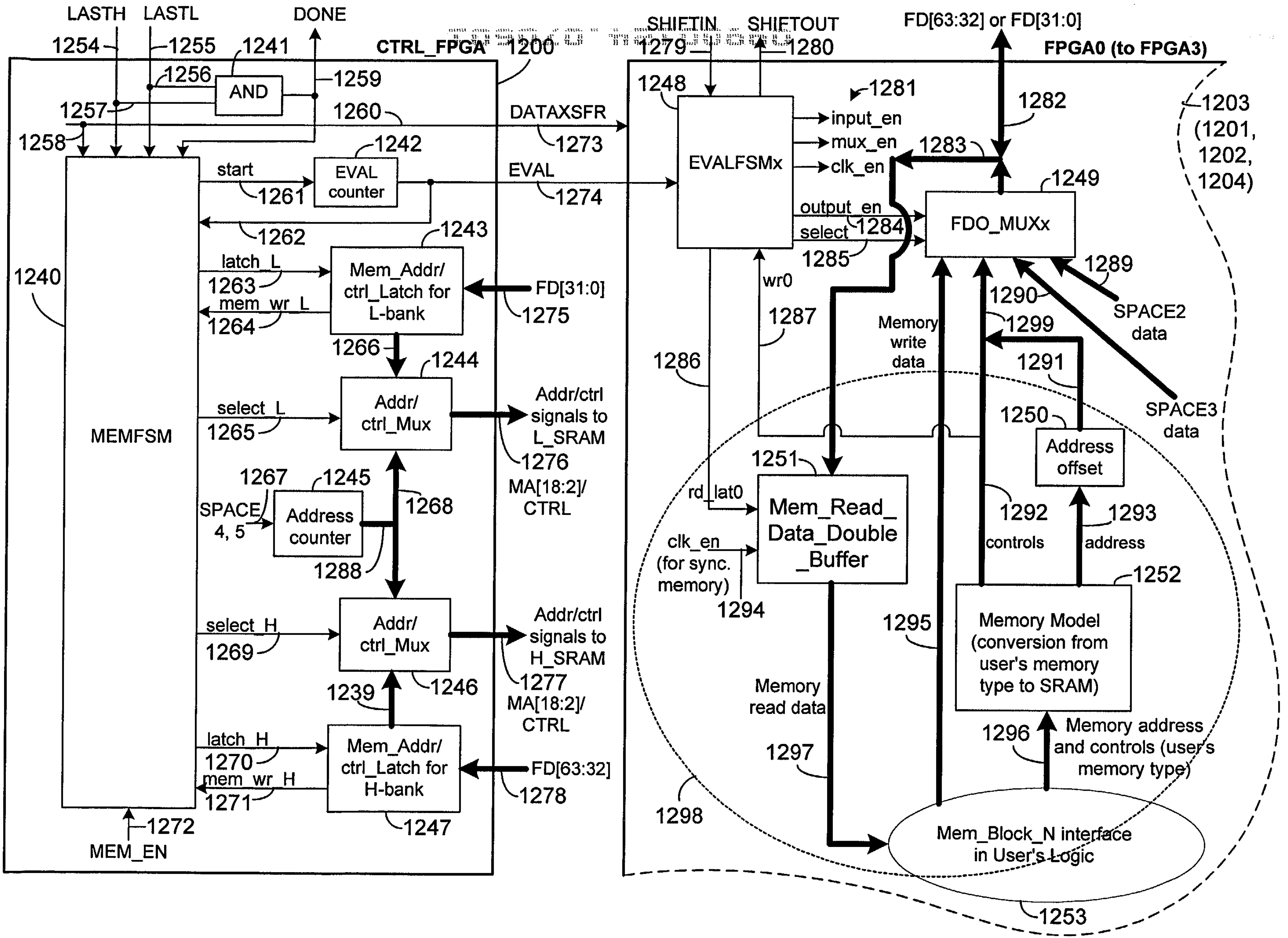
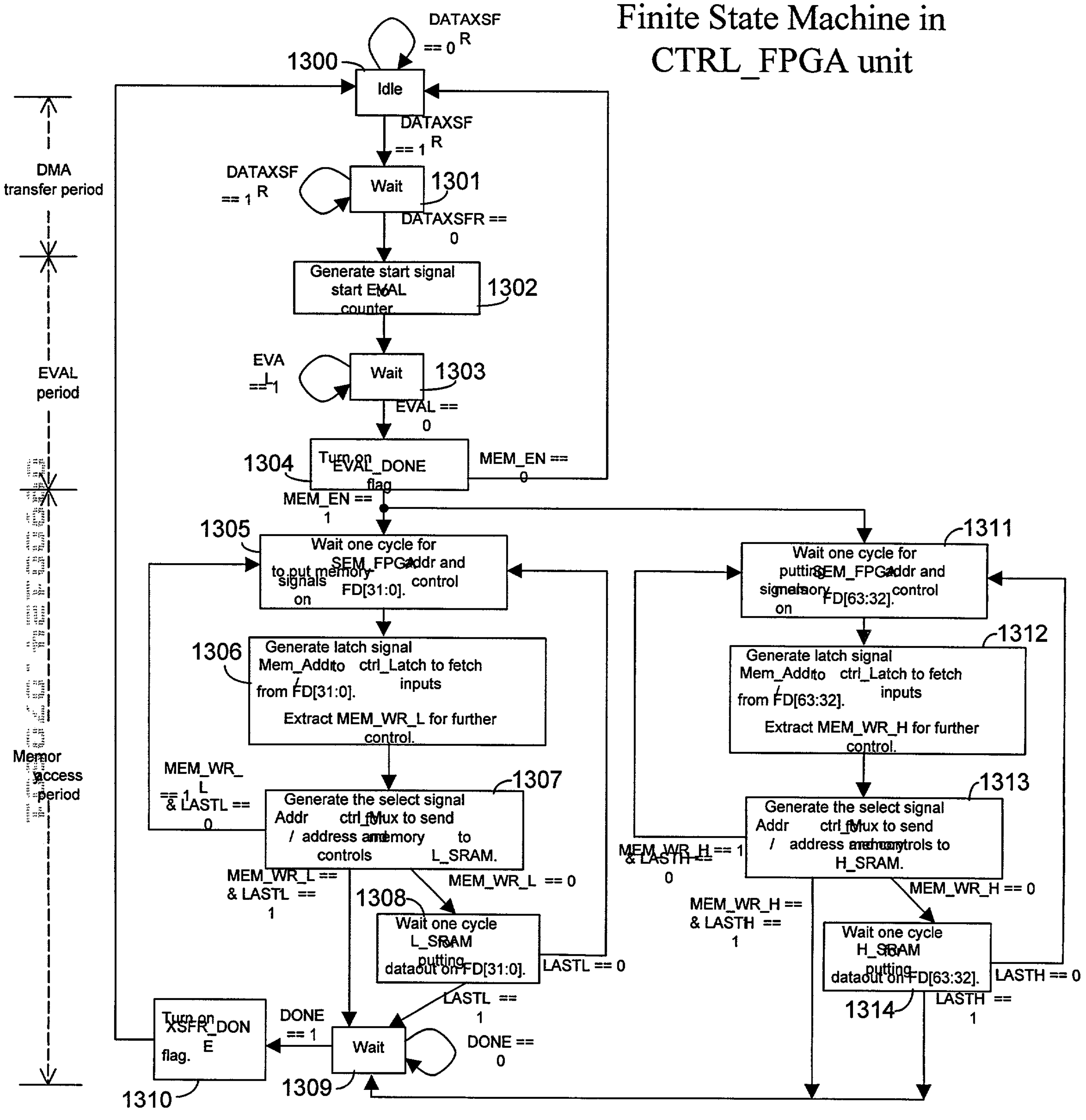

LASTH
LASTL
DONE
1254
1255
1241
1256
AND
1259
1257
1258
1260
CTRL_FPGA
1200
DATAXSFR
1273
1242
EVAL counter
start
1261
EVAL
1274
1262
1243
latch_L
1263
mem_wr_L
1264
Mem_Addr/ ctrl_Latch for L-bank
FD[31:0]
1275
1240
MEMFSM
1266
1244
select_L
1265
Addr/ ctrl_Mux
Addr/ctrl signals to L_SRAM
1276
MA[18:2]/ CTRL
1267
1245
SPACE 4, 5
Address counter
1268
1288
select_H
1269
Addr/ ctrl_Mux
Addr/ctrl signals to H_SRAM
1277
MA[18:2]/ CTRL
1239
1246
latch_H
1270
mem_wr_H
1271
Mem_Addr/ ctrl_Latch for H-bank
FD[63:32]
1278
1247
1272
MEM_EN
SHIFTIN
1279
SHIFTOUT
1280
FD[63:32] or FD[31:0]
FPGA0 (to FPGA3)
1248
EVALFSMx
1281
input_en
mux_en
clk_en
1282
1283
1203 (1201, 1202, 1204)
1249
FDO_MUXx
output_en
1284
select
1285
1289
wr0
1287
1290
1299
SPACE2 data
1286
Memory write data
1291
1250
Address offset
SPACE3 data
1251
rd_lat0
Mem_Read_ Data_Double _Buffer
clk_en (for sync. memory)
1294
1292
controls
1293
address
1252
Memory Model (conversion from user's memory type to SRAM)
1295
Memory read data
1297
1296
Memory address and controls (user's memory type)
1298
Mem_Block_N interface in User's Logic
1253

FIG. 57

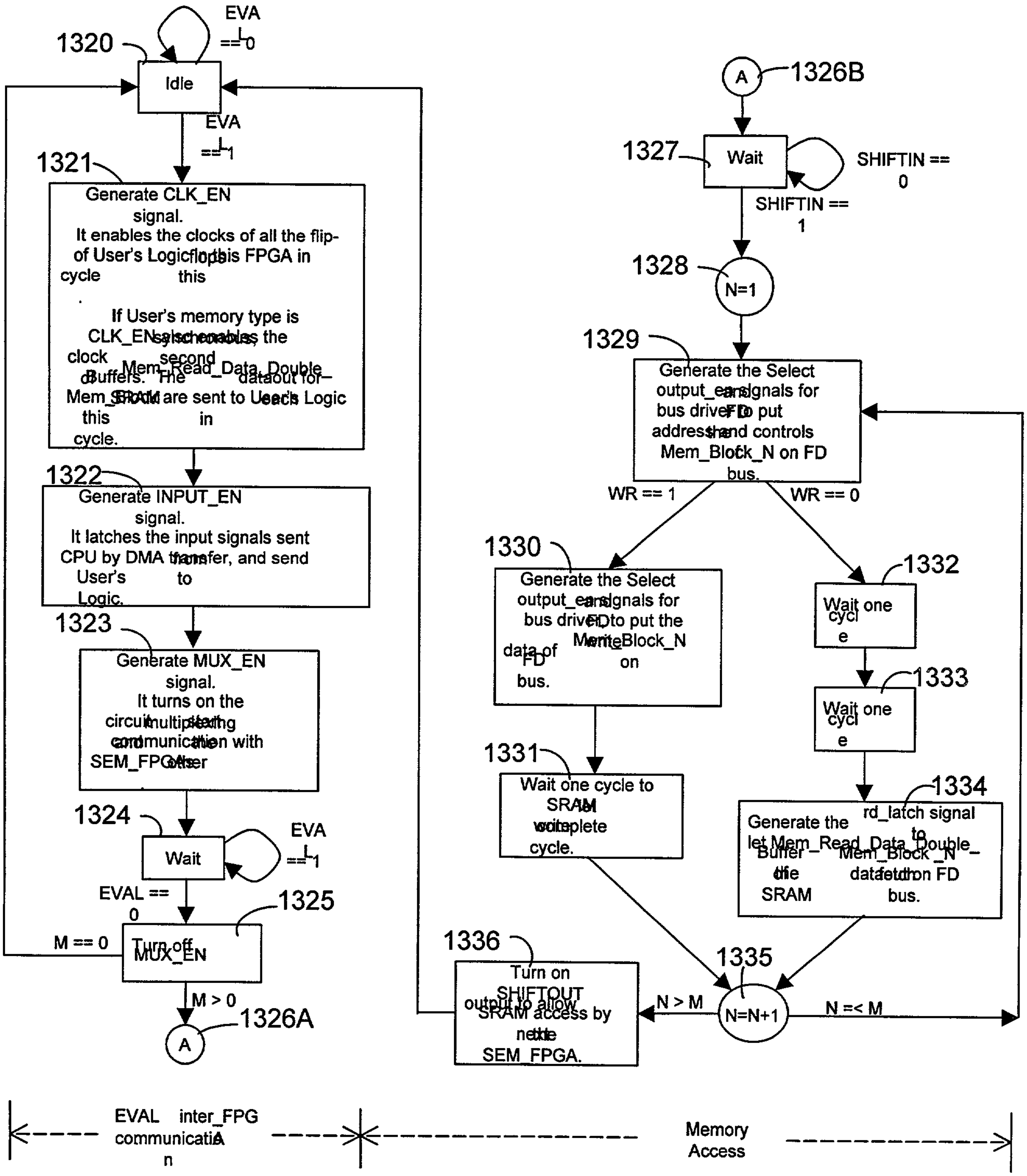


DATAXSFR == 0
1300
Idle
DATAXSFR == 1
DATAXSFR == 1
Wait
1301
DATAXSFR == 0
DMA transfer period
Generate start signal to start EVAL counter
1302
EVAL == 1
Wait
1303
EVAL == 0
EVAL period
1304
Turn on EVAL_DONE flag
MEM_EN == 0
MEM_EN == 1
1305
Wait one cycle for SEM_FPGA to put memory Addr and control signals on FD[31:0].
1311
Wait one cycle for SEM_FPGA putting memory Addr and control signals on FD[63:32].
1306
Generate latch signal to Mem_Addr / ctrl_Latch to fetch inputs from FD[31:0].
Extract MEM_WR_L for further control.
1312
Generate latch signal to Mem_Addr / ctrl_Latch to fetch inputs from FD[63:32].
Extract MEM_WR_H for further control.
Memory access period
MEM_WR_L == 1 & LASTL == 0
1307
Generate the select signal for Addr / ctrl_Mux to send address and memory controls to L_SRAM.
1313
Generate the select signal for Addr / ctrl_Mux to send address and memory controls to H_SRAM.
MEM_WR_H == 1 & LASTH == 0
MEM_WR_L == 1 & LASTL == 1
MEM_WR_L == 0
1308
Wait one cycle for L_SRAM putting dataout on FD[31:0].
LASTL == 0
MEM_WR_H == 1 & LASTH == 1
MEM_WR_H == 0
Wait one cycle for H_SRAM putting dataout on FD[63:32].
LASTH == 0
1314
LASTH == 1
LASTL == 1
Turn on XSFR_DONE flag.
DONE == 1
Wait
DONE == 0
1310
1309

MEMFSM - Memory Finite State Machine in CTRL_FPGA unit

FIG. 58

# EVALFSM - EVAL
# Finite State Machine in
# each FPGA logic device


1320
Idle
EVA ==0
EVA ==1
1321
Generate CLK_EN signal.
It enables the clocks of all the flip-flops of User's Logic in this FPGA in this cycle
.
If User's memory type is synchronous, CLK_EN also enables the second clock Mem_Read_Data_Double_Buffers. The dataout for Mem_Block are sent to User's Logic in this cycle.
1322
Generate INPUT_EN signal.
It latches the input signals sent from CPU by DMA transfer, and send to User's Logic.
1323
Generate MUX_EN signal.
It turns on the multiplexing circuit starting communication with other SEM_FPGAs.
1324
Wait
EVA ==1
EVAL == 0
1325
Turn off MUX_EN
M == 0
M > 0
1326A
A
A
1326B
1327
Wait
SHIFTIN == 0
SHIFTIN == 1
1328
N=1
1329
Generate the Select and output_en signals for bus driver to put the address and controls Mem_Block_N on FD bus.
WR == 1
WR == 0
1330
Generate the Select and output_en signals for FD bus driver to put the data of Mem_Block_N on FD bus.
Write
1331
Wait one cycle to let SRAM complete write cycle.
1332
Wait one cycle
1333
Wait one cycle
1334
Generate the rd_latch signal to let Mem_Read_Data_Double_Buffer_Mem_Block_N to latch the data fetched from SRAM on FD bus.
1335
N=N+1
N > M
N =< M
1336
Turn on SHIFTOUT output to allow SRAM access by the next SEM_FPGA.
EVAL inter_FPGA communication
Memory Access

# FIG. 59

# MEMORY READ DATA DOUBLE BUFFER

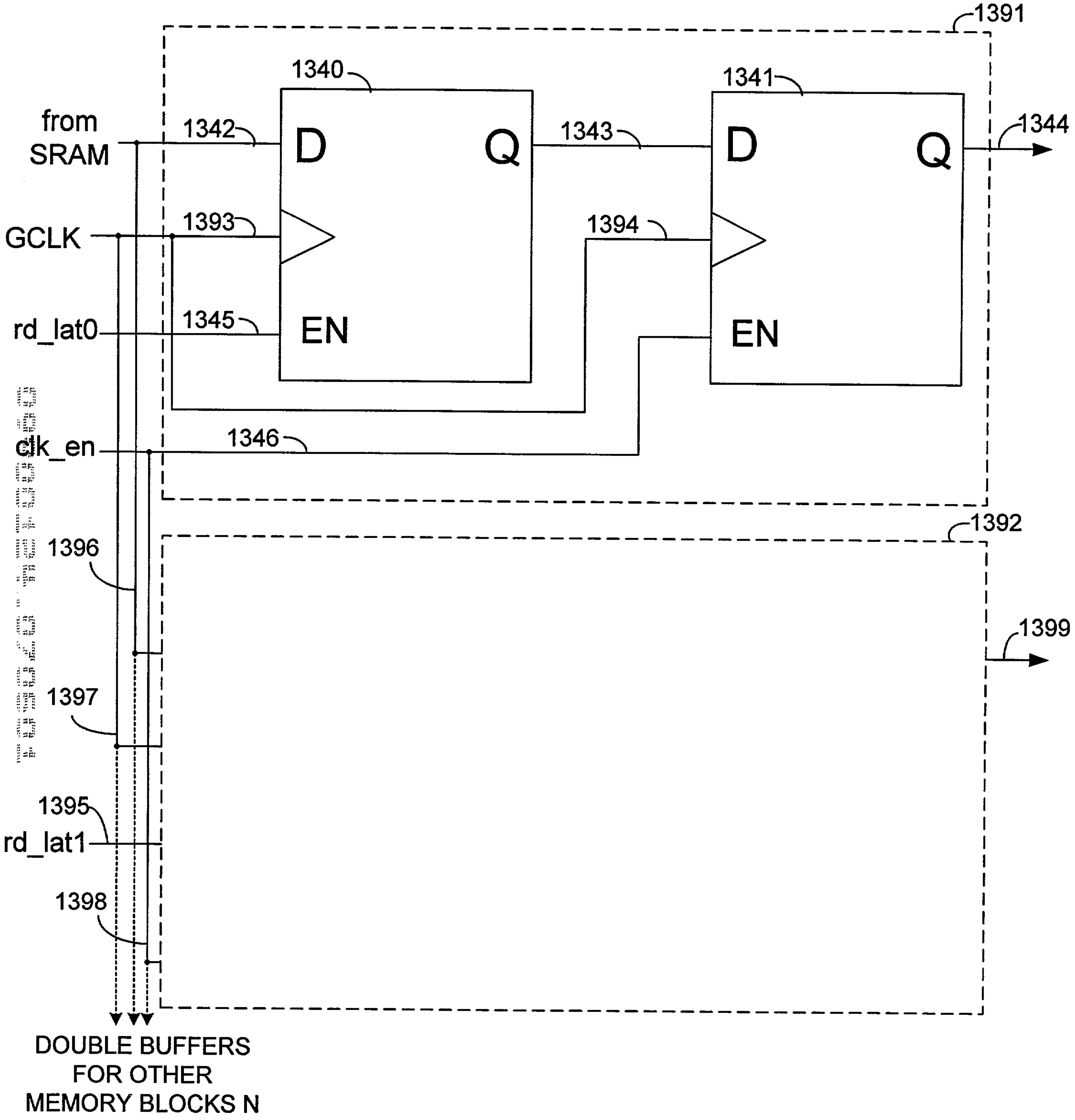


FIG. 60

SIMULATION WRITE/READ CYCLE

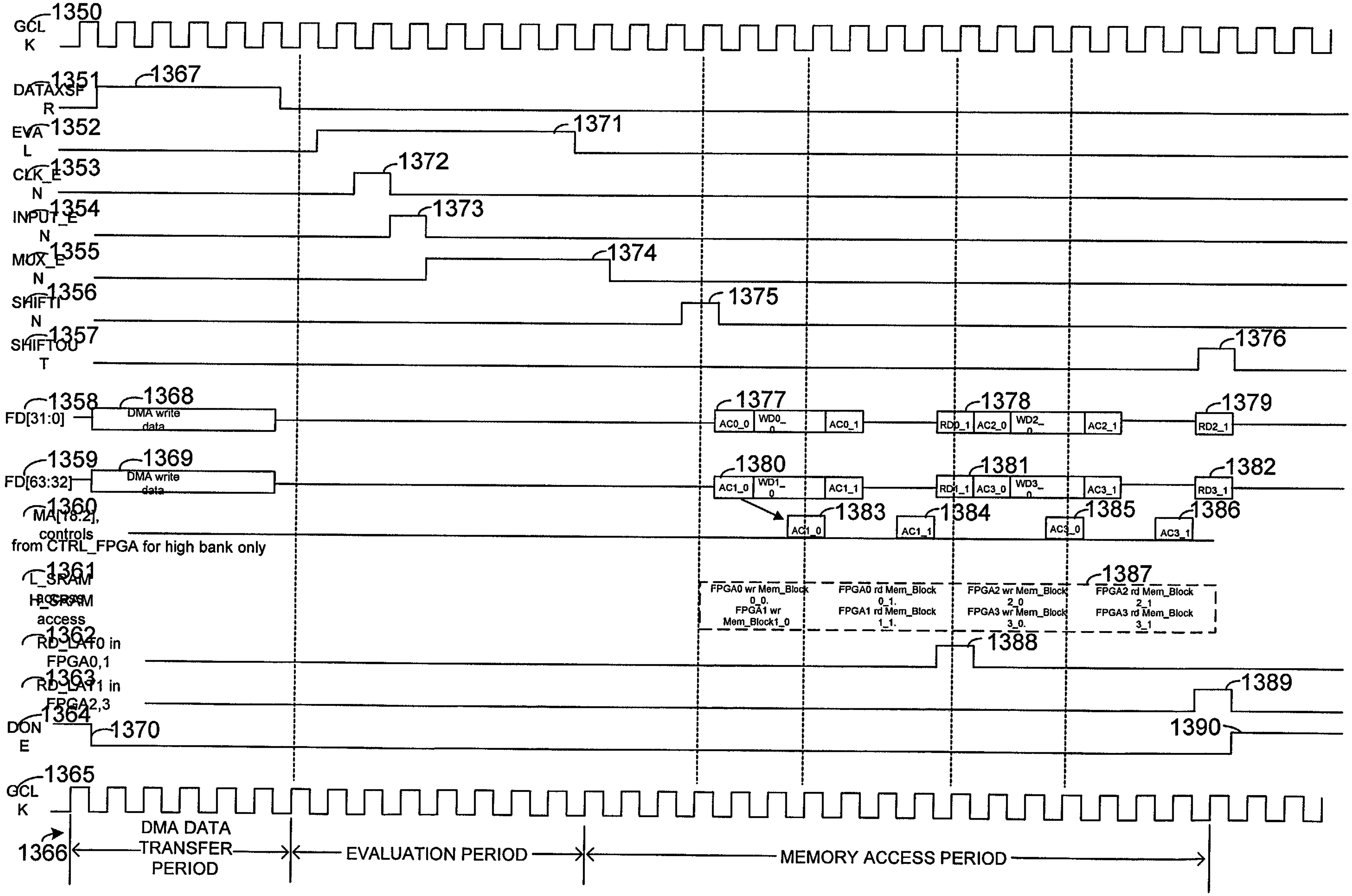


FIG. 61

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=0)

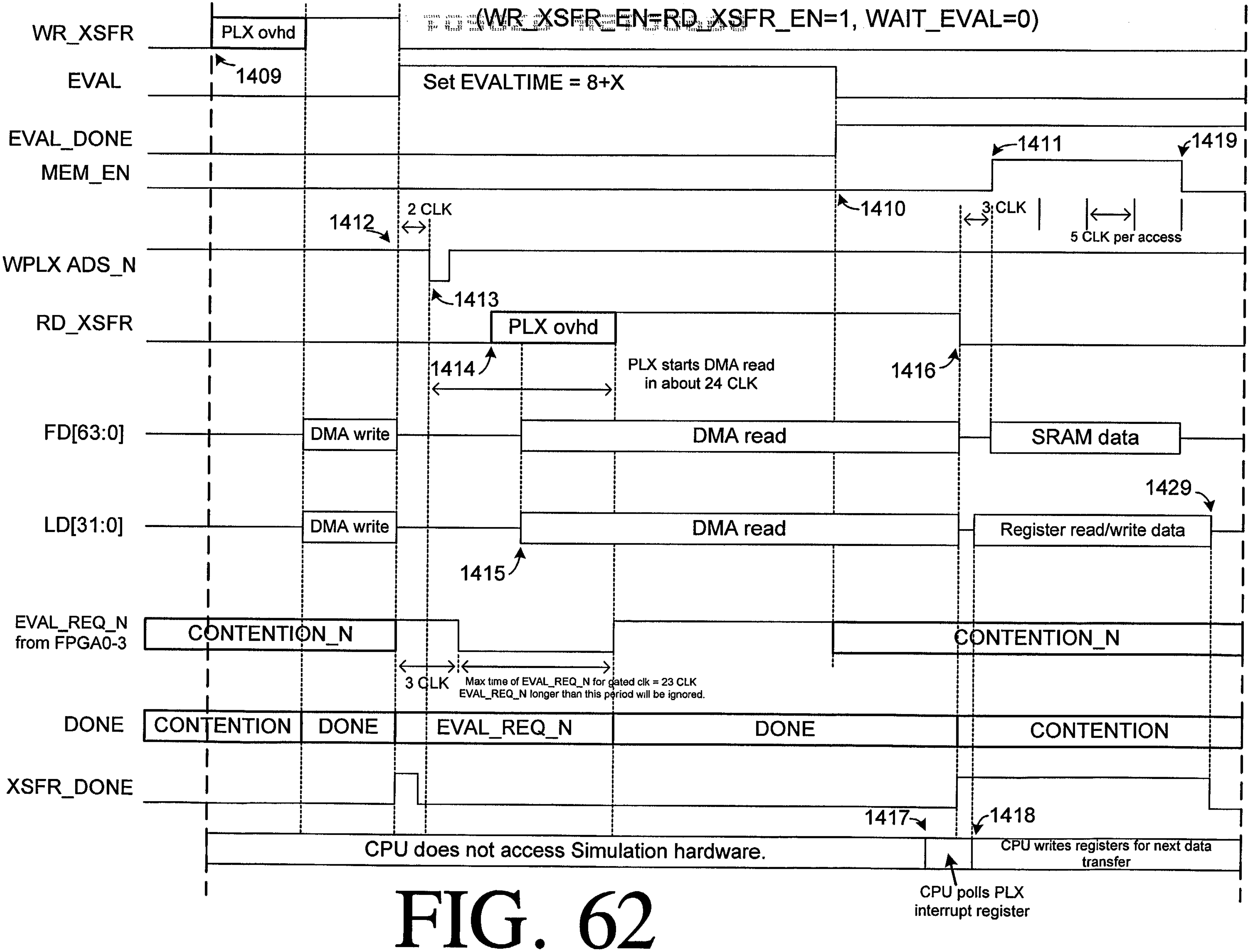


FIG. 62

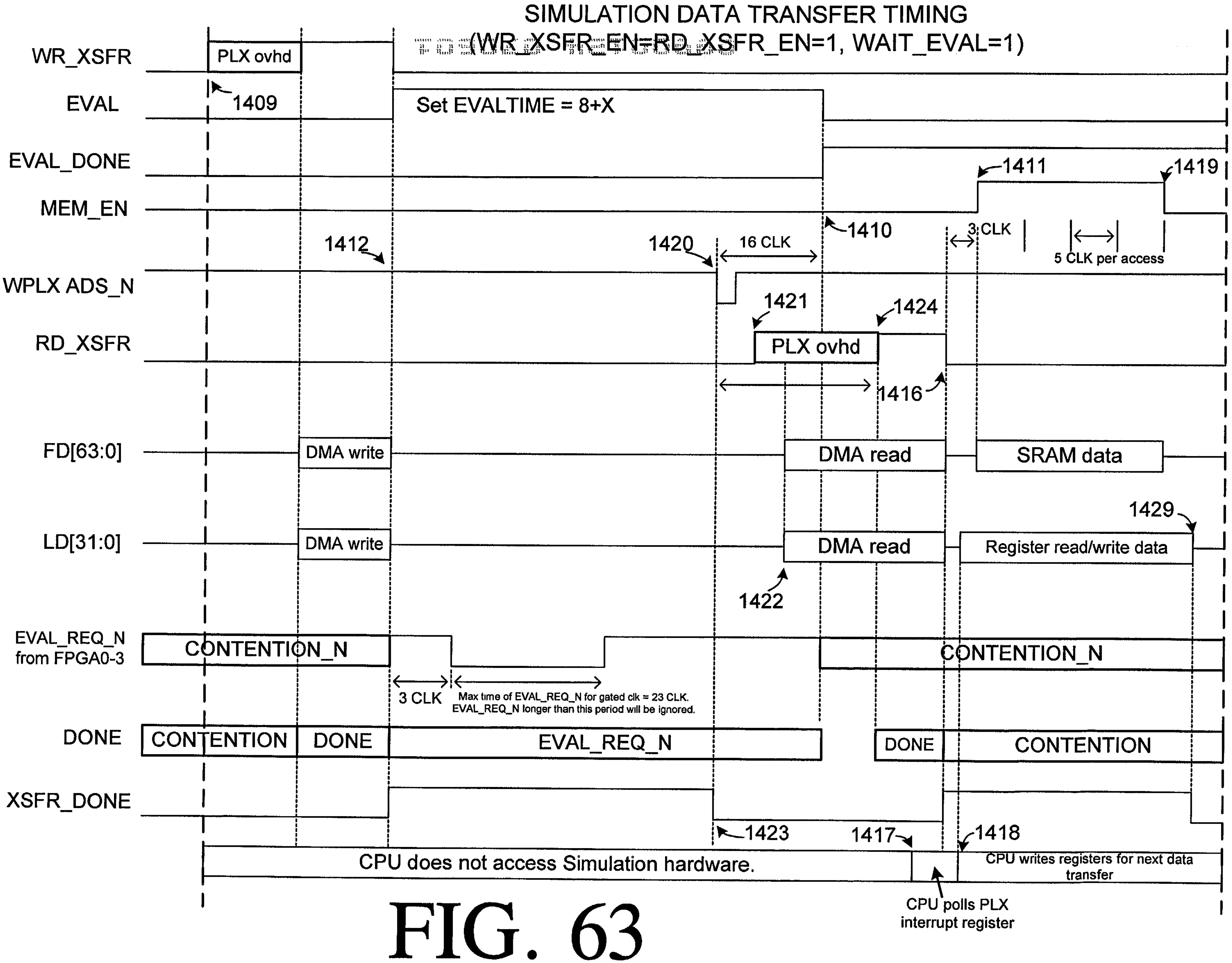

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=1)
WR_XSFR
PLX ovhd
EVAL
1409
Set EVALTIME = 8+X
EVAL_DONE
1411
1419
MEM_EN
1410
3 CLK
5 CLK per access
1412
1420
16 CLK
WPLX ADS_N
1421
1424
RD_XSFR
PLX ovhd
1416
FD[63:0]
DMA write
DMA read
SRAM data
1429
LD[31:0]
DMA write
DMA read
Register read/write data
1422
EVAL_REQ_N
from FPGA0-3
CONTENTION_N
CONTENTION_N
3 CLK
Max time of EVAL_REQ_N for gated clk = 23 CLK.
EVAL_REQ_N longer than this period will be ignored.
DONE
CONTENTION
DONE
EVAL_REQ_N
DONE
CONTENTION
XSFR_DONE
1423
1417
1418
CPU does not access Simulation hardware.
CPU writes registers for next data transfer
CPU polls PLX interrupt register

FIG. 63

# Typical User Design of PCI Add-on Cards

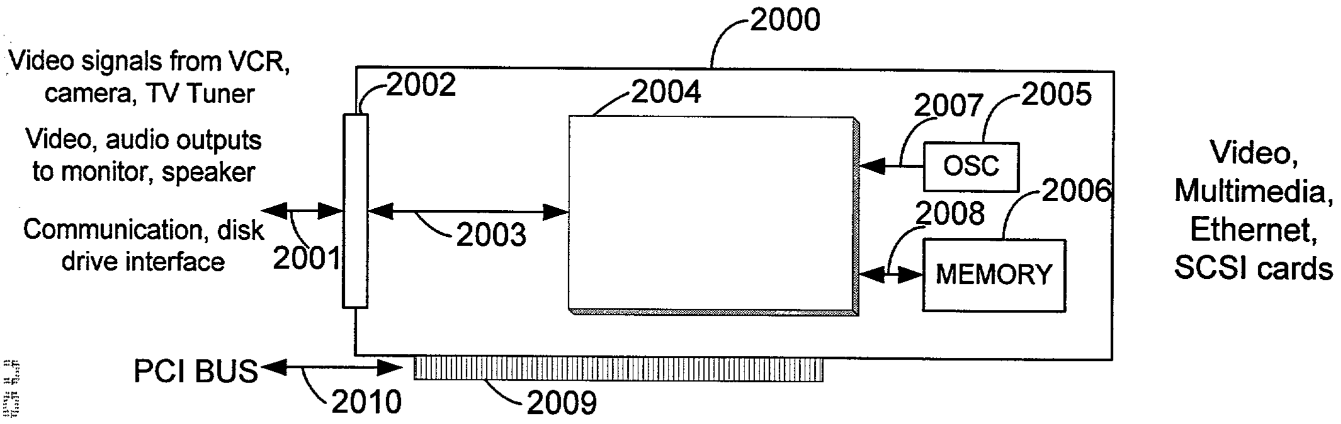

2000
Video signals from VCR, camera, TV Tuner
Video, audio outputs to monitor, speaker
Communication, disk drive interface
2001
2002
2003
2004
2007
2005
OSC
2006
2008
MEMORY
Video, Multimedia, Ethernet, SCSI cards
PCI BUS
2010
2009

FIG. 64

# Typical Hardware/Software Co-Verification

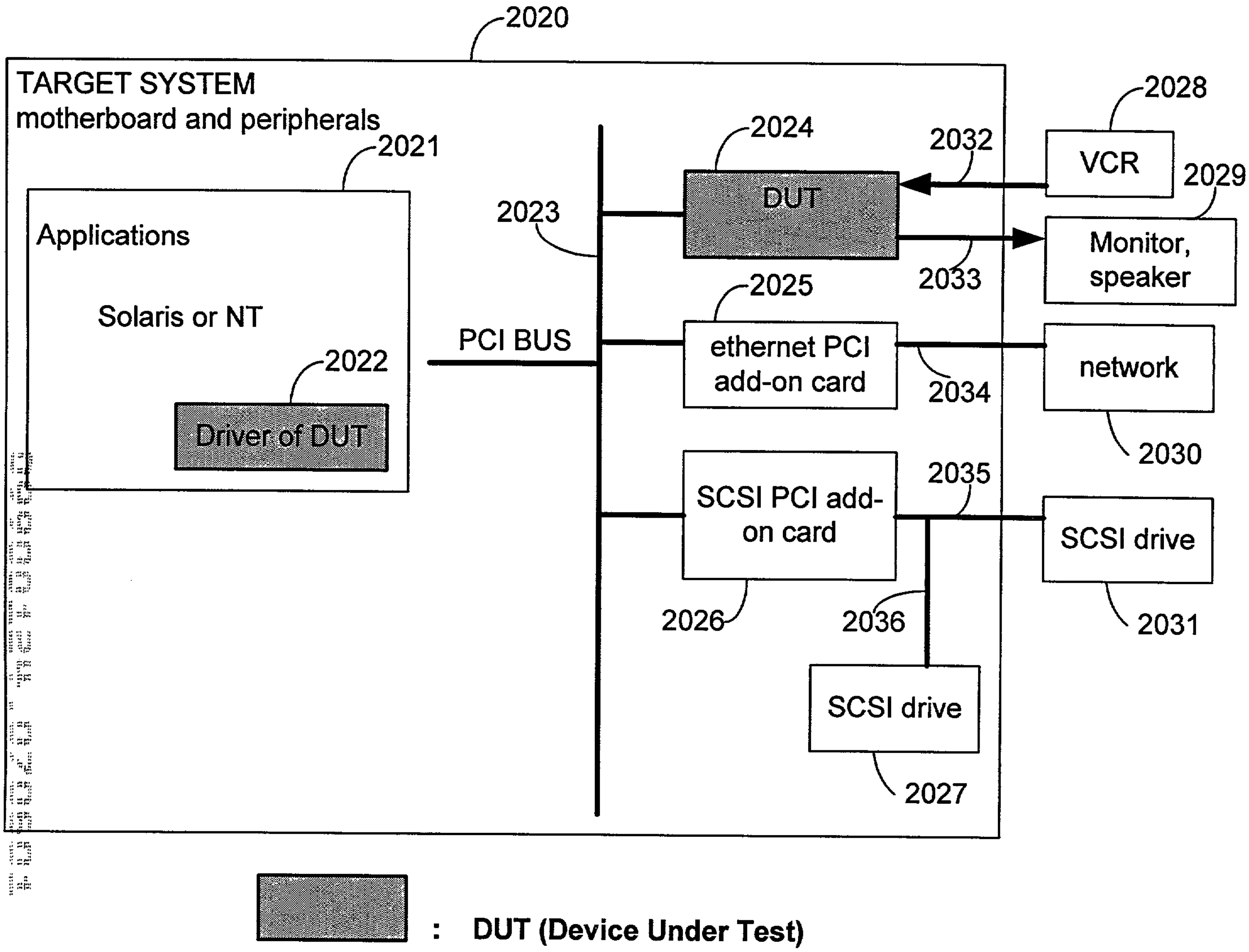
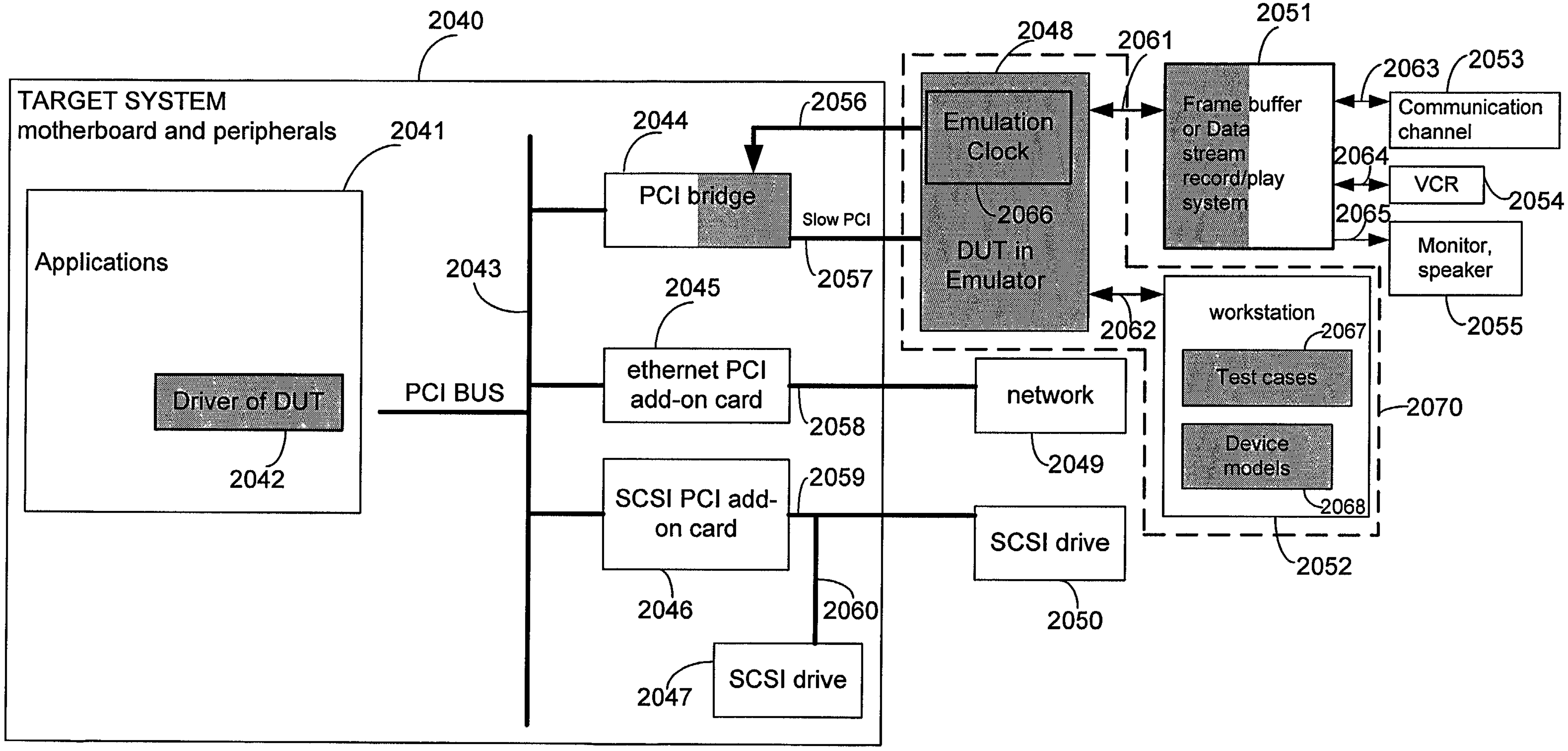


FIG. 65

# Typical Co-Verification by Using Emulator



▨ : running time at emulation speed

The rest of the target system is running at full speed.

FIG. 66

# SIMULATION

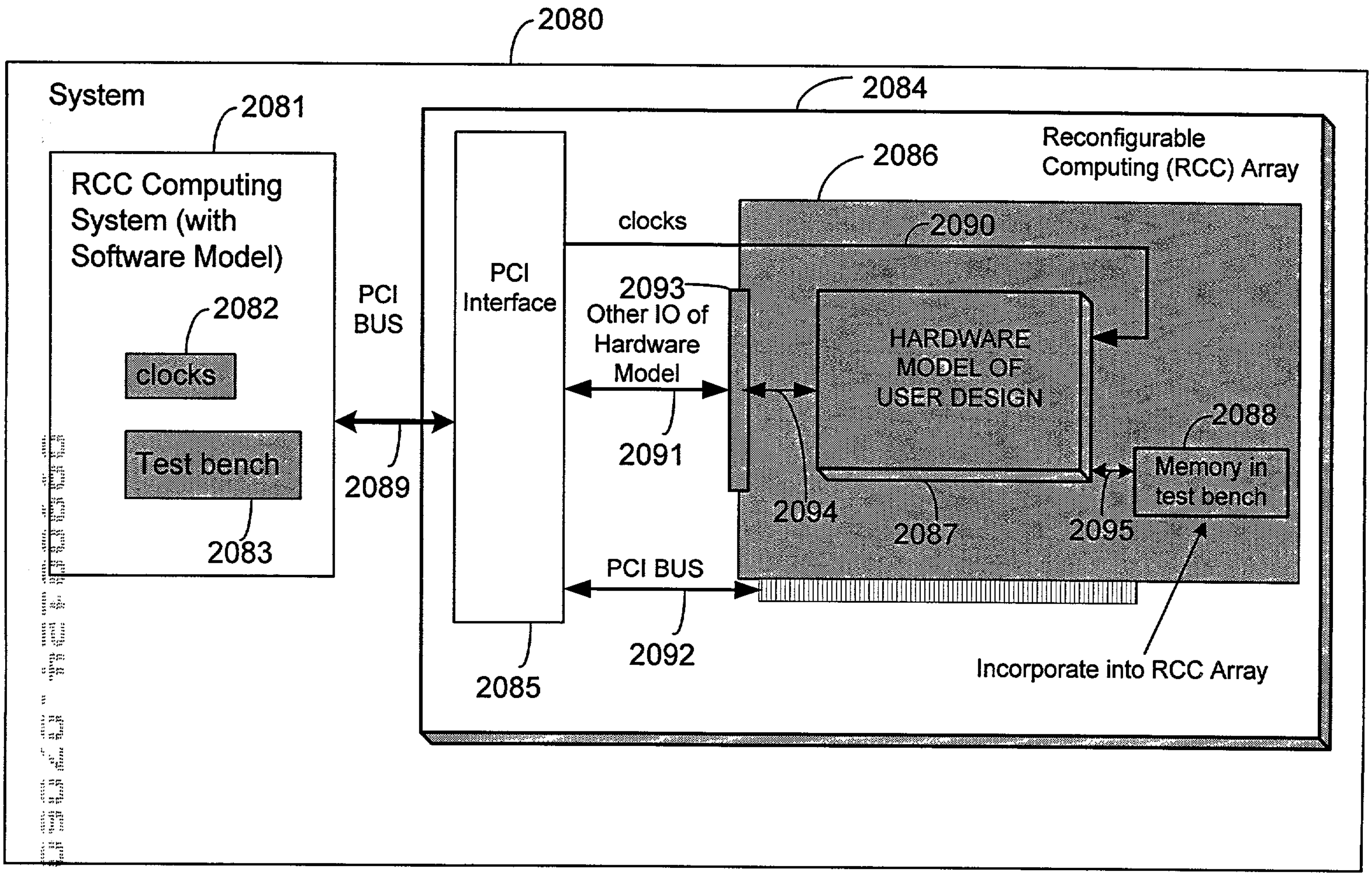

2080
System
2081
RCC Computing System (with Software Model)
2082
clocks
Test bench
2083
PCI BUS
2089
2084
Reconfigurable Computing (RCC) Array
PCI Interface
2085
clocks
2086
2090
2093
Other IO of Hardware Model
2091
HARDWARE MODEL OF USER DESIGN
2094
2087
2088
Memory in test bench
2095
PCI BUS
2092
Incorporate into RCC Array

FIG. 67

# CO-VERIFICATION WITHOUT EXTERNAL I/O

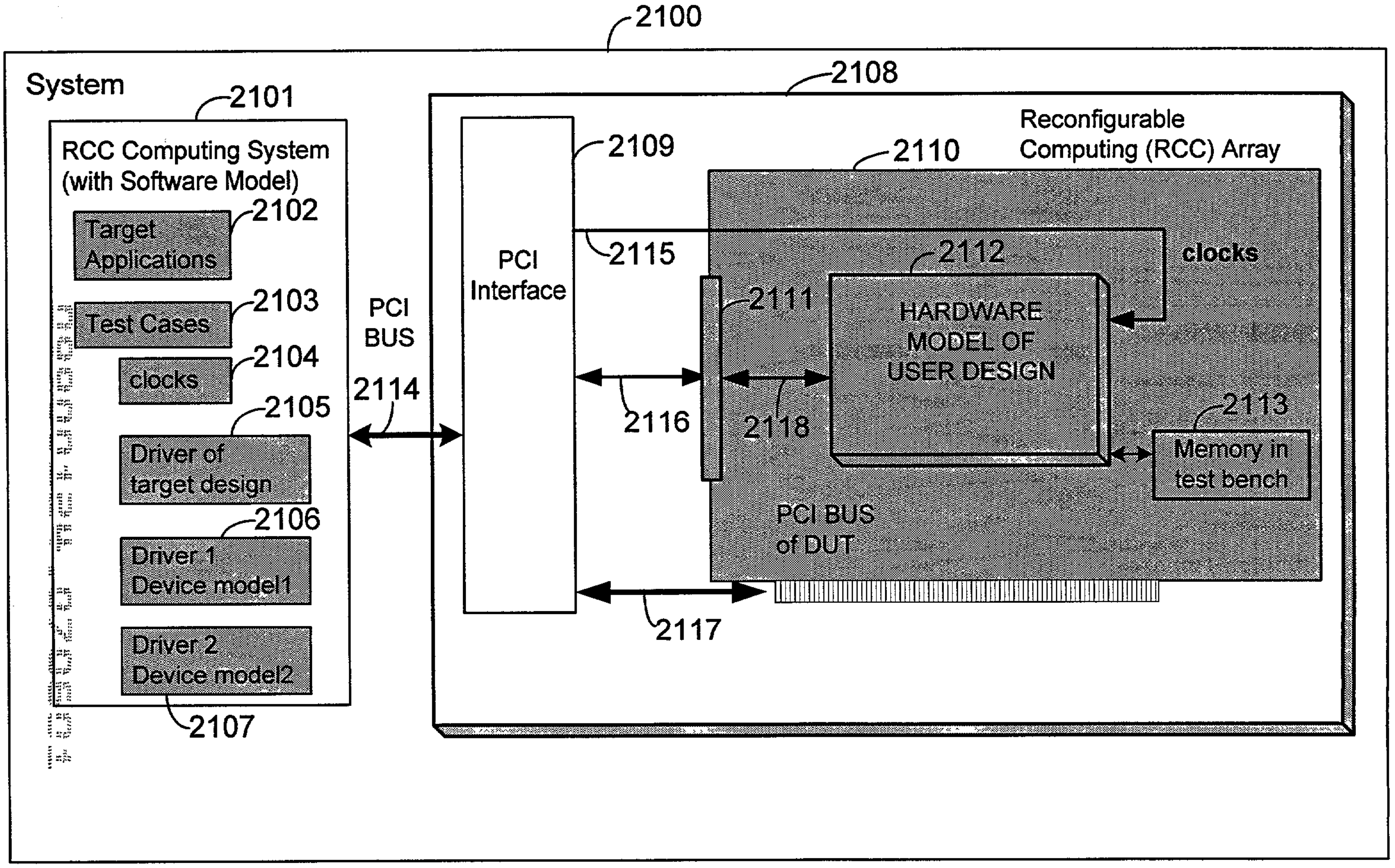


FIG. 68

# CO-VERIFICATION WITH EXTERNAL I/O

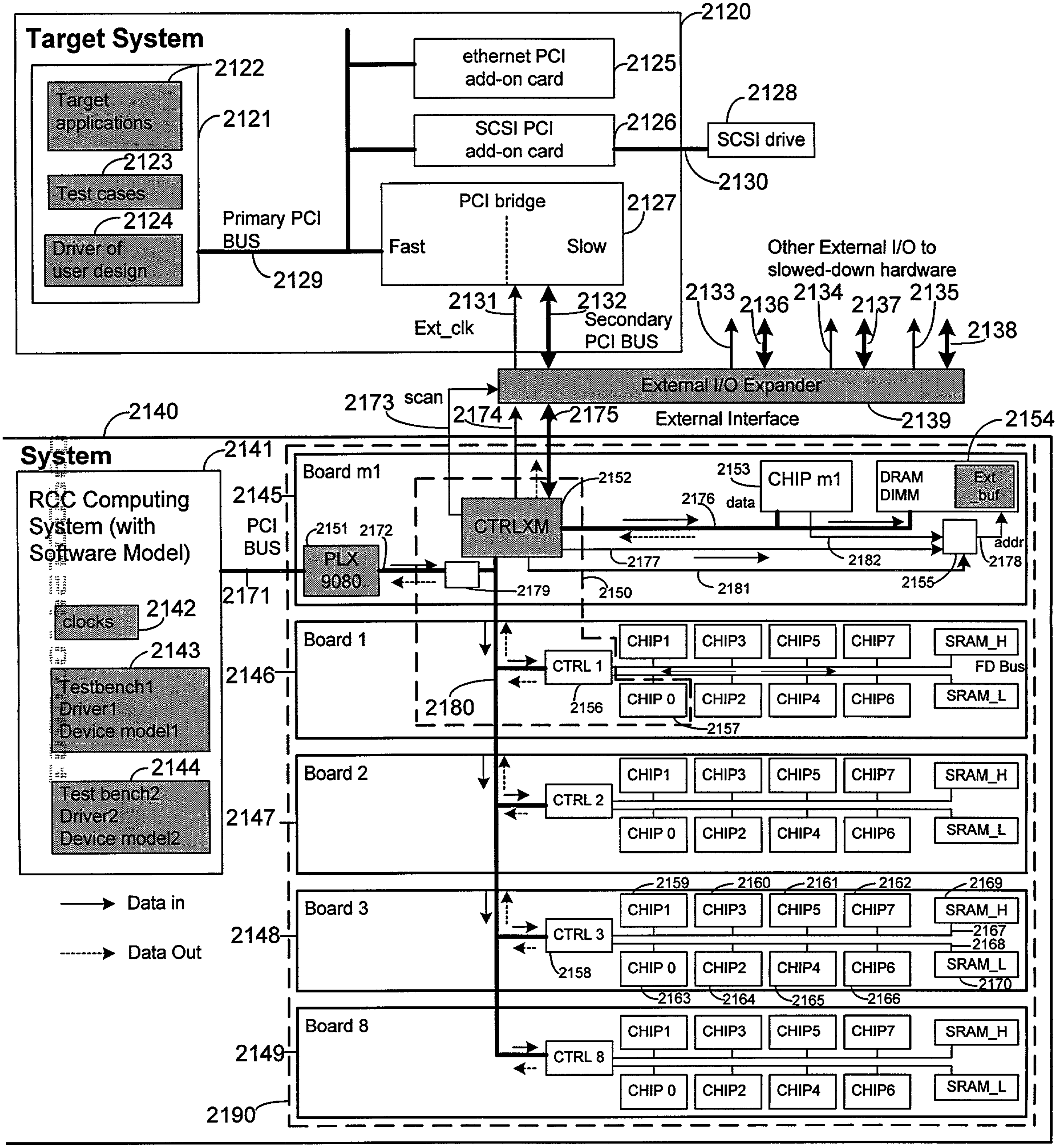


FIG. 69

# CONTROL OF DATA-IN CYCLE

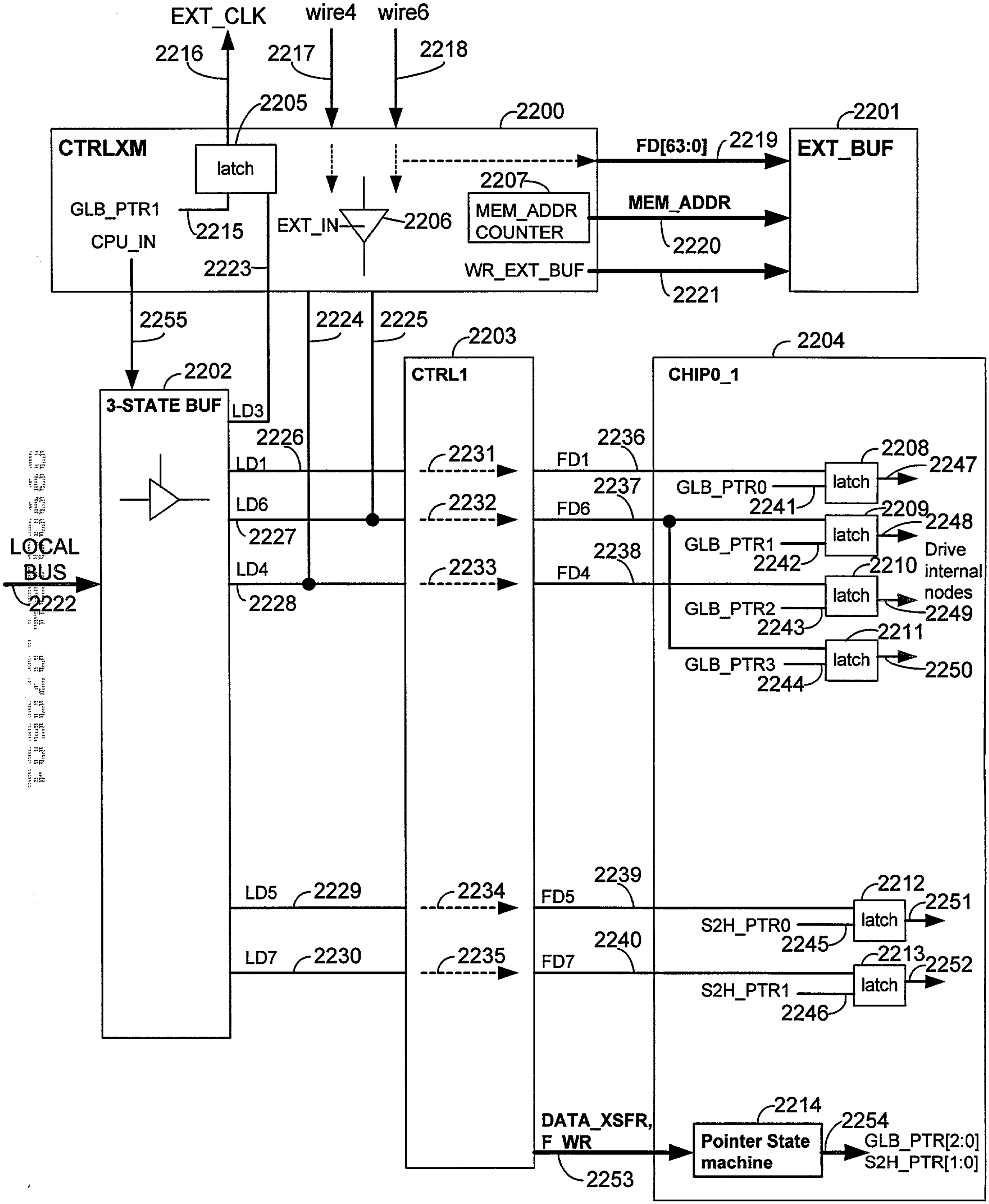


FIG. 70

# CONTROL OF DATA-OUT CYCLE

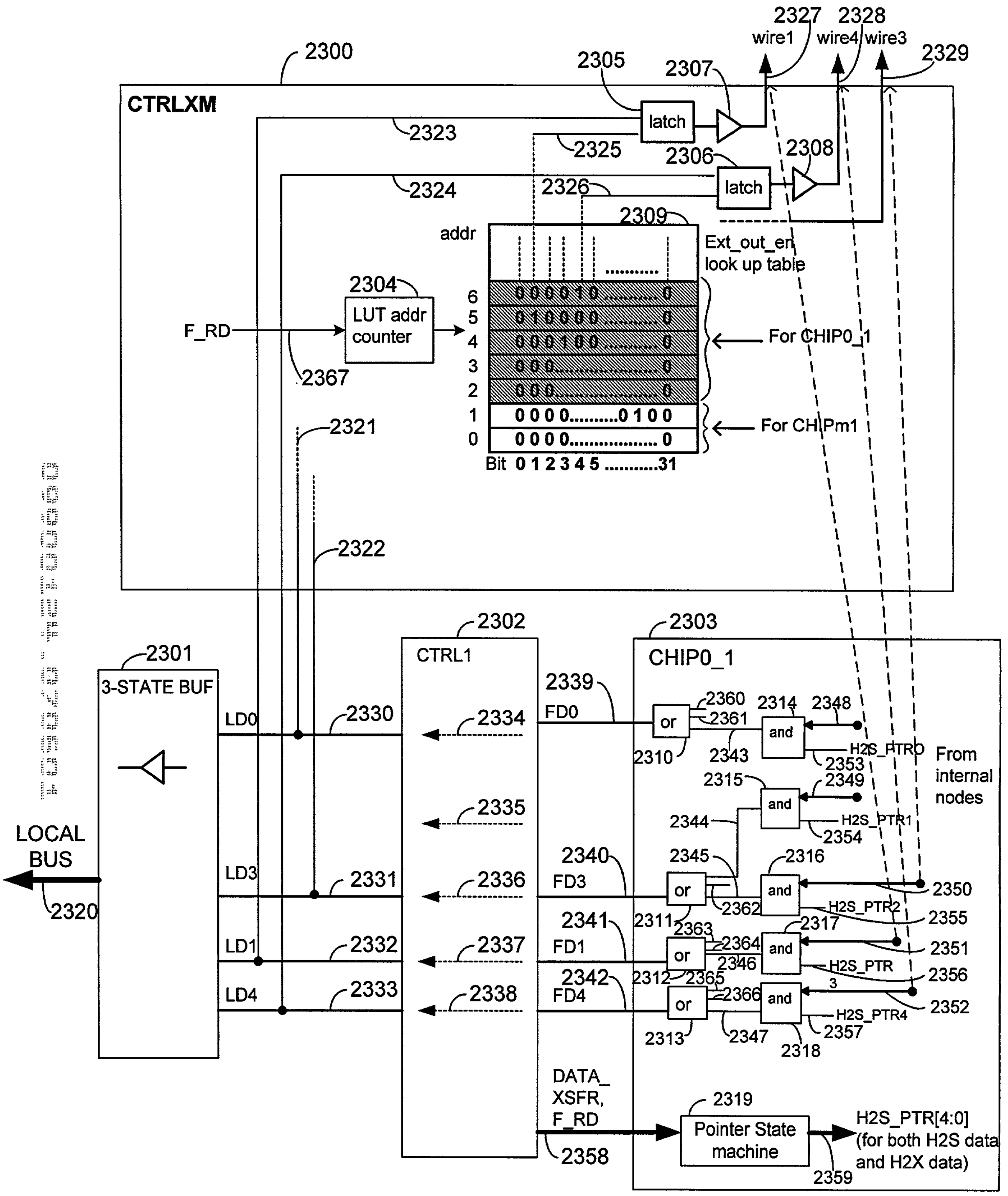


FIG. 71

# CONTROL OF DATA-IN CYCLE

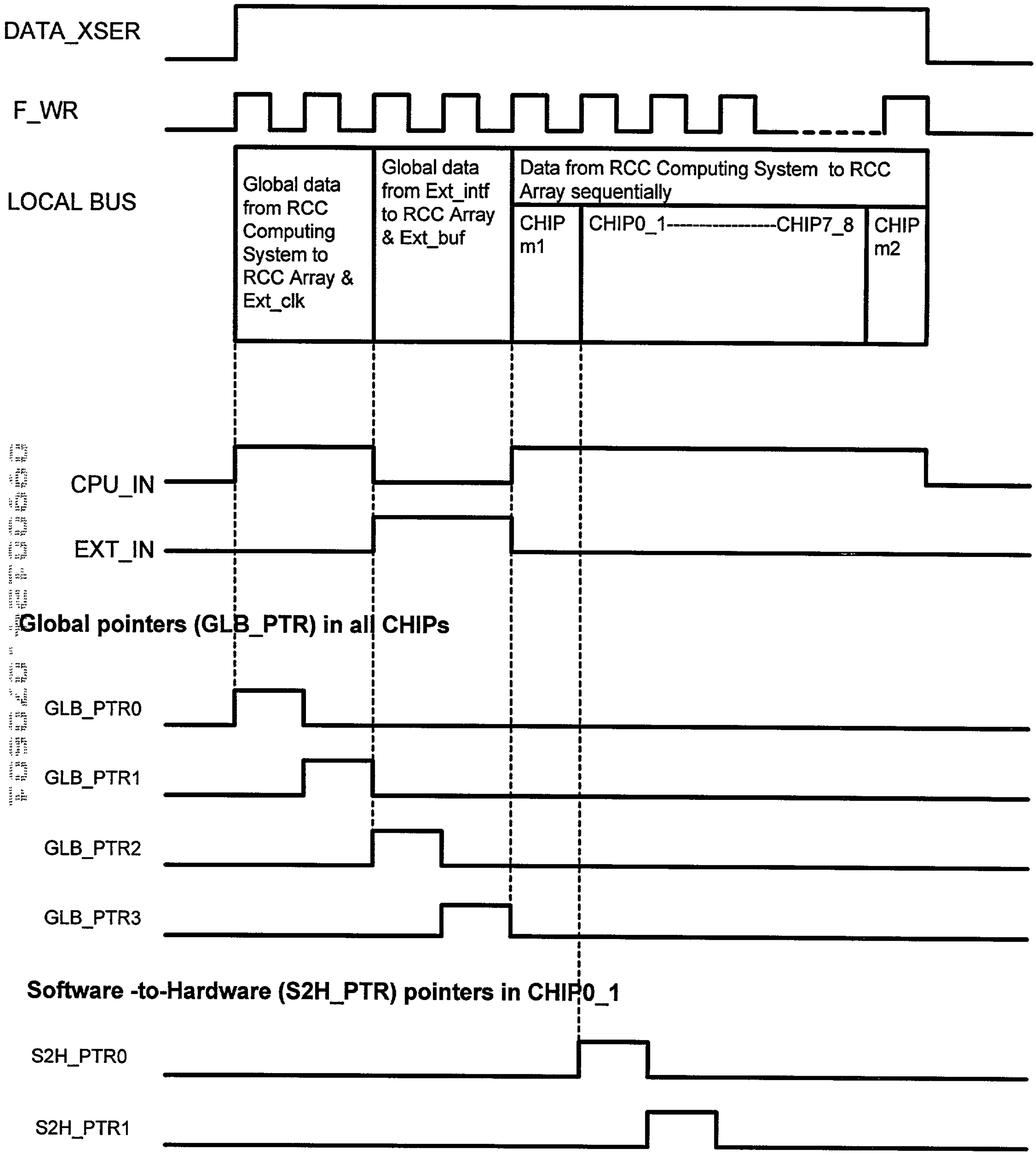

DATA_XSER
F_WR
LOCAL BUS
Global data from RCC Computing System to RCC Array & Ext_clk
Global data from Ext_intf to RCC Array & Ext_buf
Data from RCC Computing System to RCC Array sequentially
CHIP m1
CHIP0_1----------------CHIP7_8
CHIP m2
CPU_IN
EXT_IN
Global pointers (GLB_PTR) in all CHIPs
GLB_PTR0
GLB_PTR1
GLB_PTR2
GLB_PTR3
Software -to-Hardware (S2H_PTR) pointers in CHIP0_1
S2H_PTR0
S2H_PTR1

FIG. 72

# CONTROL OF DATA-OUT CYCLE

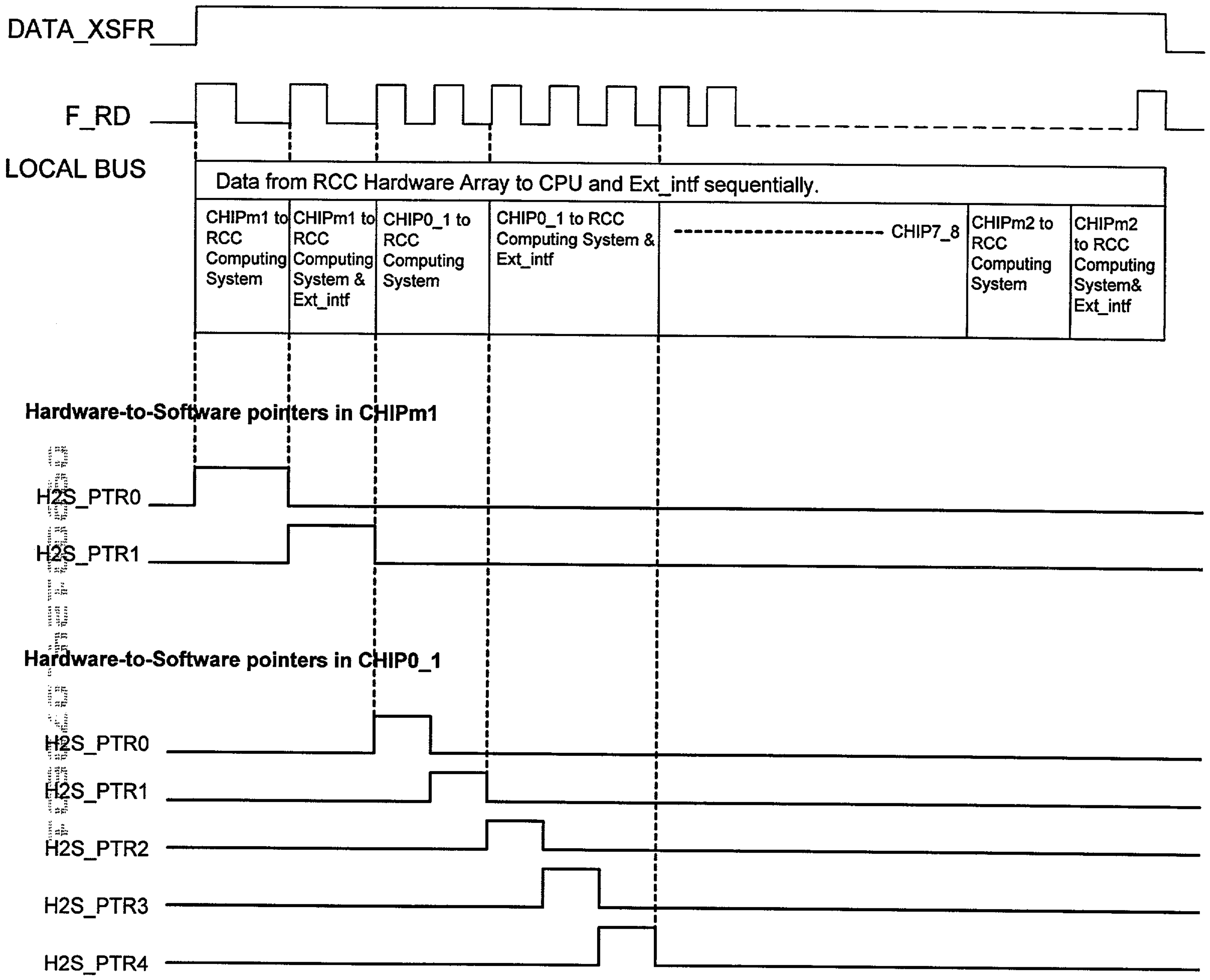
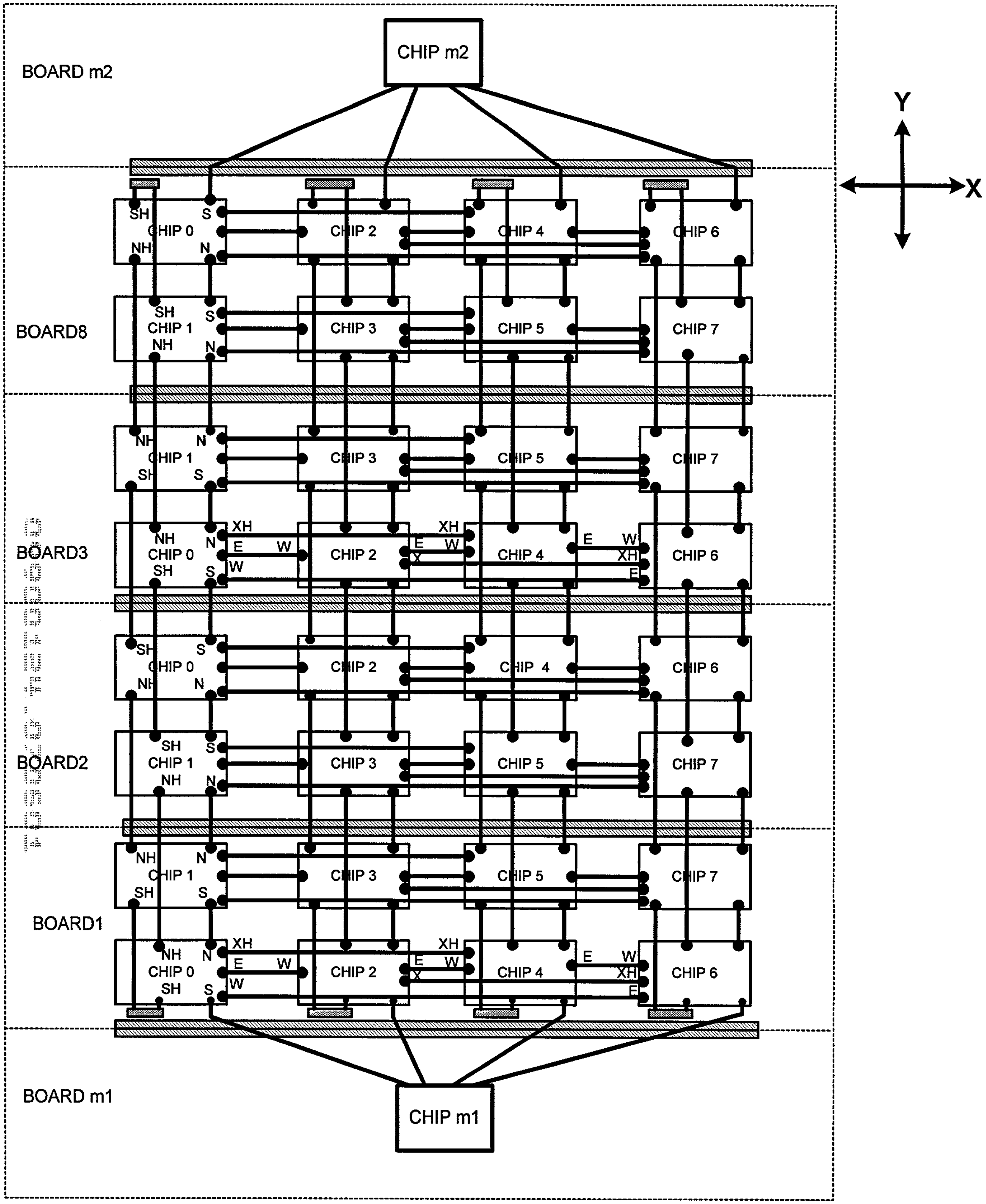

DATA_XSFR
F_RD
LOCAL BUS
Data from RCC Hardware Array to CPU and Ext_intf sequentially.
CHIPm1 to RCC Computing System
CHIPm1 to RCC Computing System & Ext_intf
CHIP0_1 to RCC Computing System
CHIP0_1 to RCC Computing System & Ext_intf
CHIP7_8
CHIPm2 to RCC Computing System
CHIPm2 to RCC Computing System& Ext_intf
Hardware-to-Software pointers in CHIPm1
H2S_PTR0
H2S_PTR1
Hardware-to-Software pointers in CHIP0_1
H2S_PTR0
H2S_PTR1
H2S_PTR2
H2S_PTR3
H2S_PTR4

FIG. 73



BOARD m2
CHIP m2
Y
X
SH
S
CHIP 0
NH
N
CHIP 2
CHIP 4
CHIP 6
BOARD8
SH
CHIP 1
NH
S
N
CHIP 3
CHIP 5
CHIP 7
NH
N
CHIP 1
SH
S
CHIP 3
CHIP 5
CHIP 7
BOARD3
NH
N
CHIP 0
SH
S
XH
E
W
W
CHIP 2
XH
E
W
X
CHIP 4
E
W
XH
E
CHIP 6
SH
S
CHIP 0
NH
N
CHIP 2
CHIP 4
CHIP 6
BOARD2
SH
S
CHIP 1
NH
N
CHIP 3
CHIP 5
CHIP 7
NH
N
CHIP 1
SH
S
CHIP 3
CHIP 5
CHIP 7
BOARD1
NH
N
CHIP 0
SH
S
XH
E
W
W
CHIP 2
XH
E
W
X
CHIP 4
E
W
XH
E
CHIP 6
BOARD m1
CHIP m1

FIG. 74

# SHIFT REGISTER

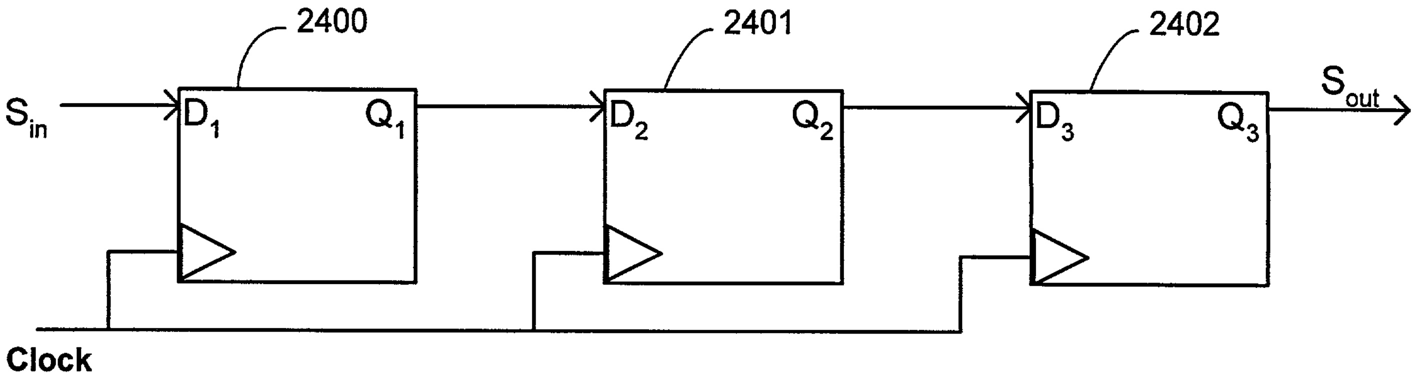


FIG. 75(A)

# HOLD TIME ASSUMPTION FOR SHIFT REGISTER

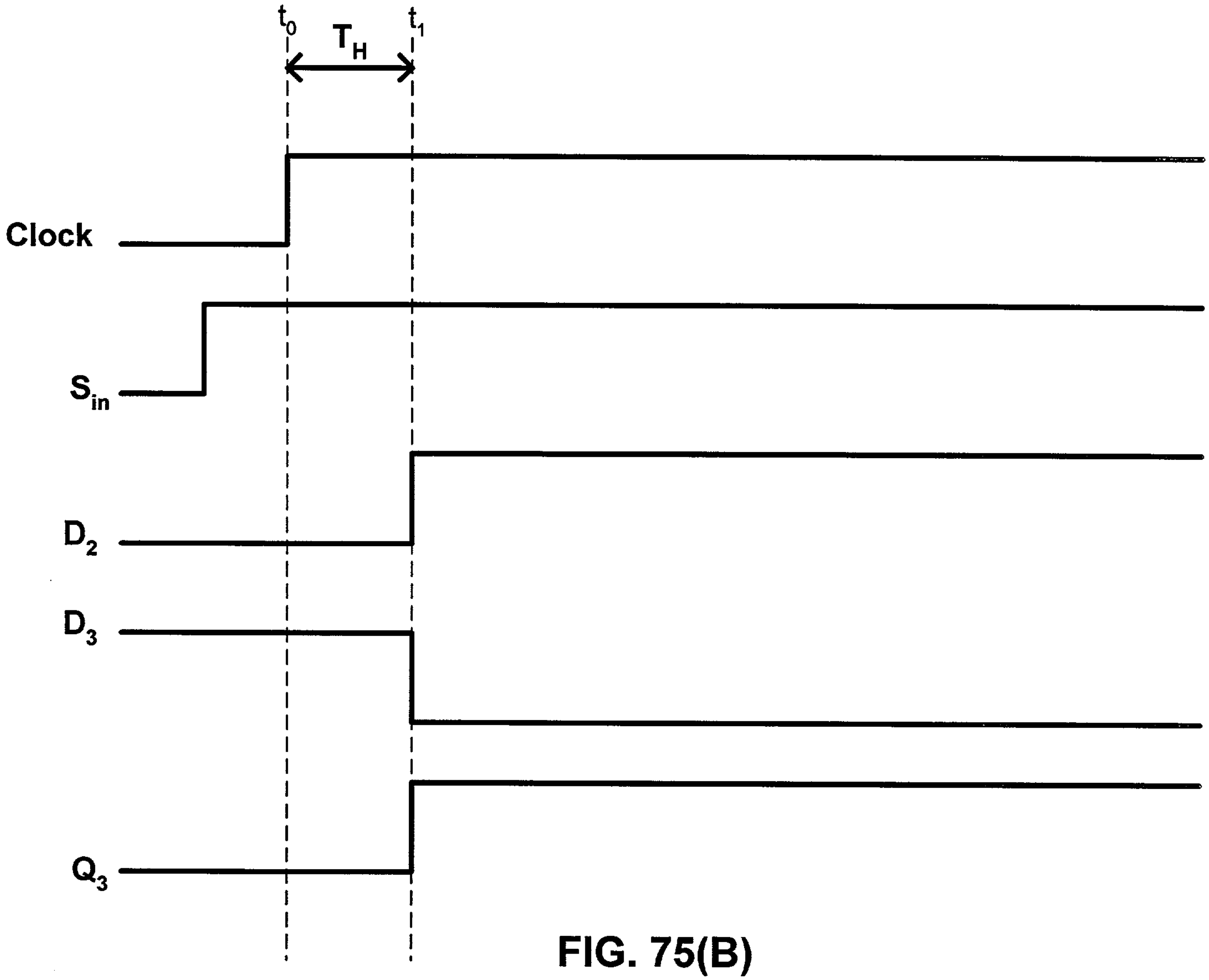


FIG. 75(B)

## MULTIPLE FPGA MAPPING FOR SHIFT REGISTER

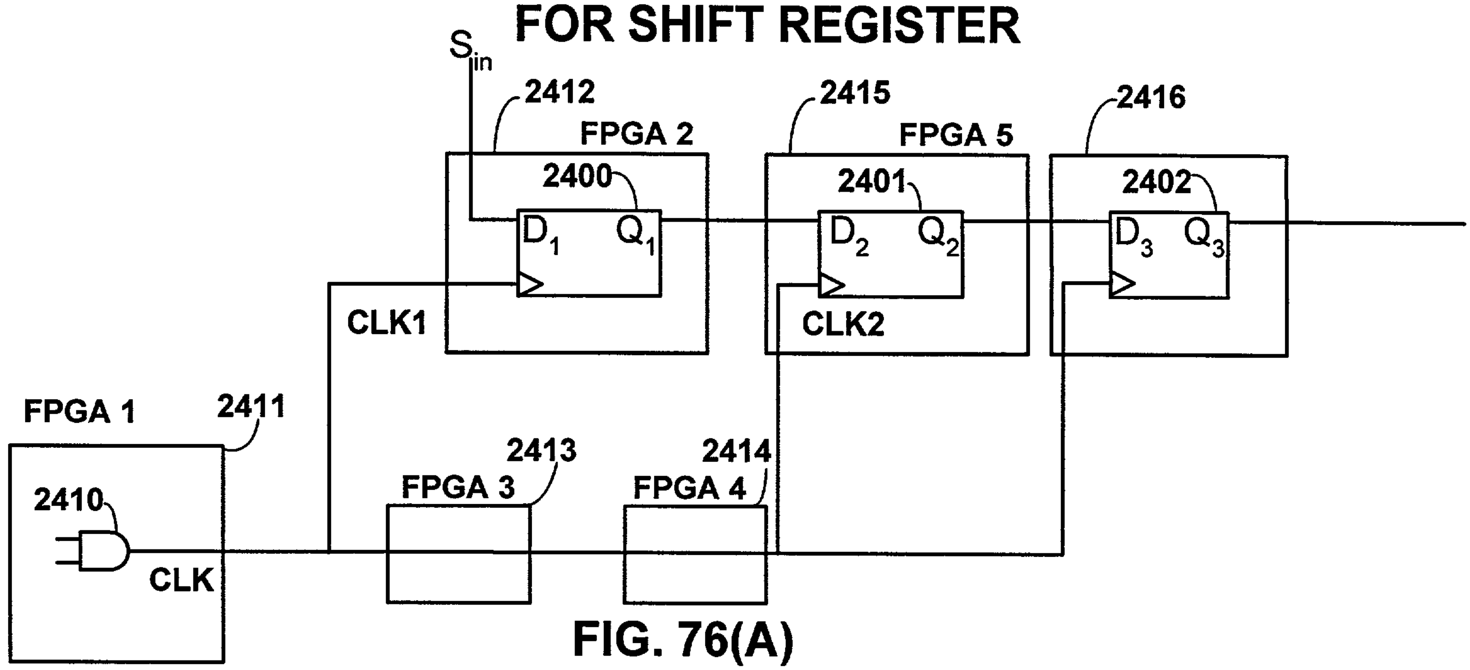


FIG. 76(A)

## HOLD TIME VIOLATION BY LONG CLOCK SKEW

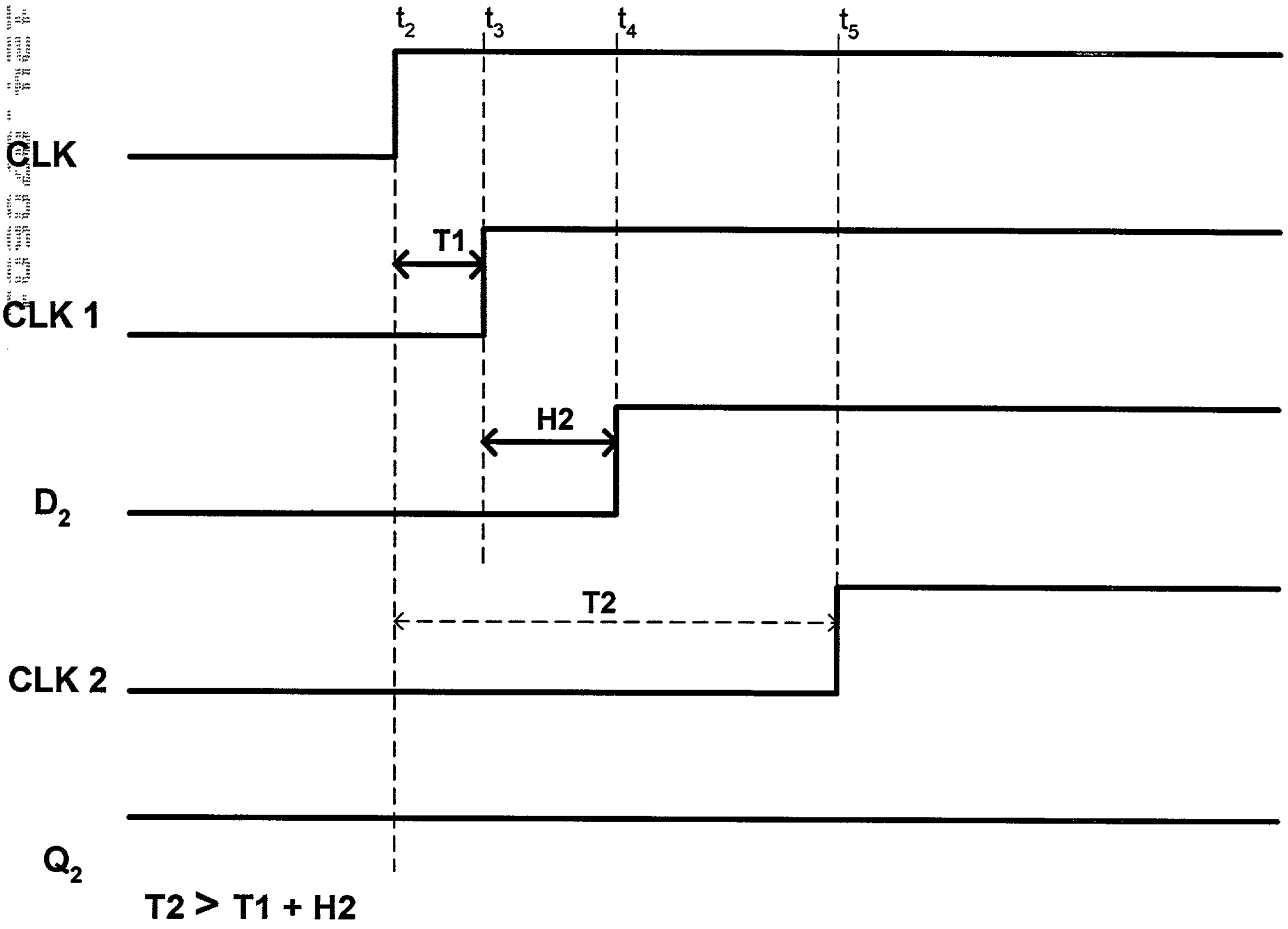


FIG. 76(B)

# CLOCK GLITCH PROBLEM

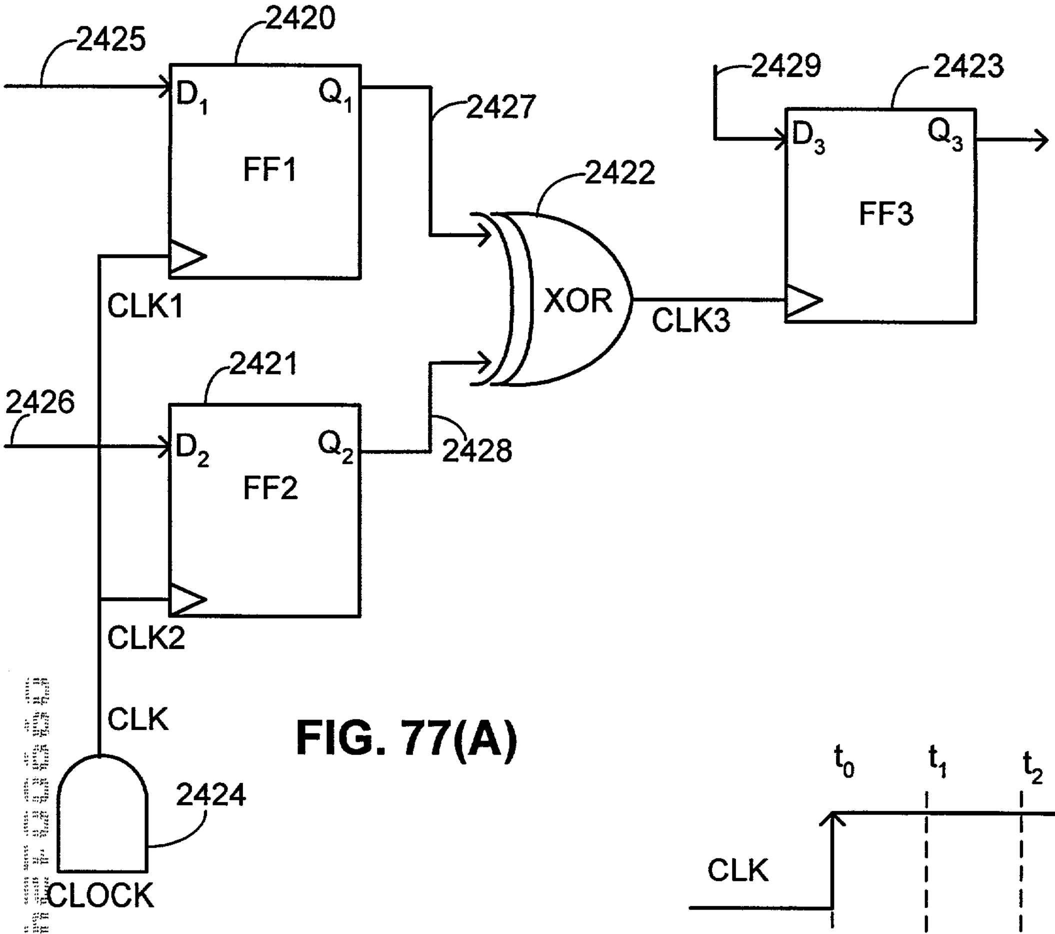


FIG. 77(A)

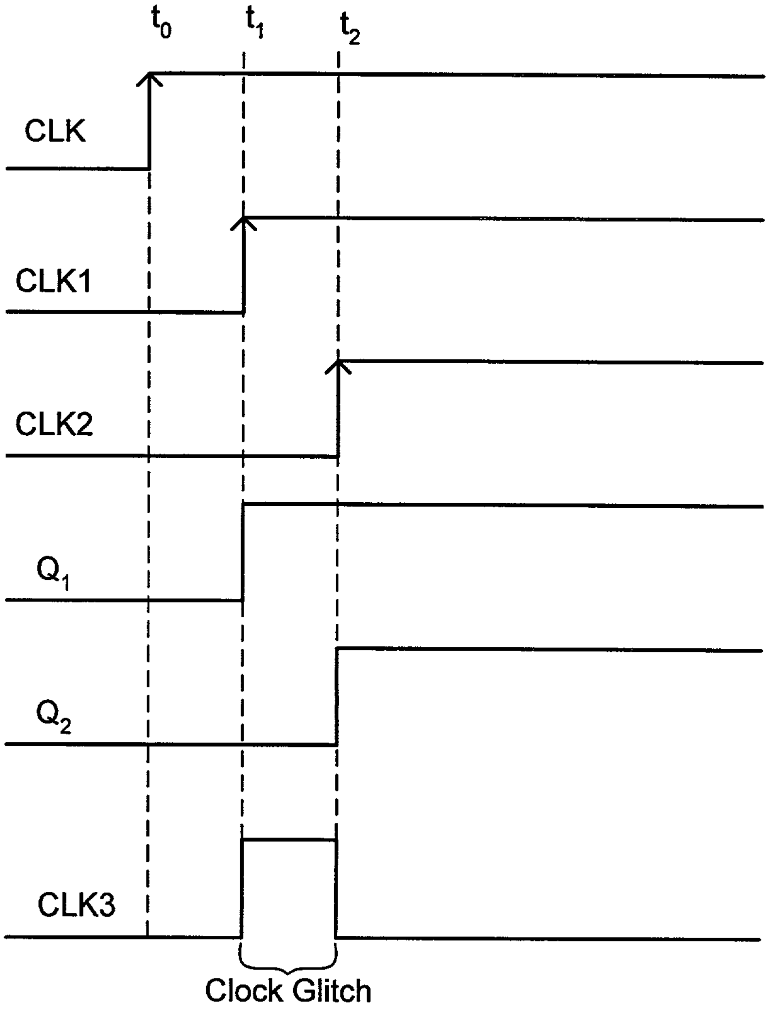
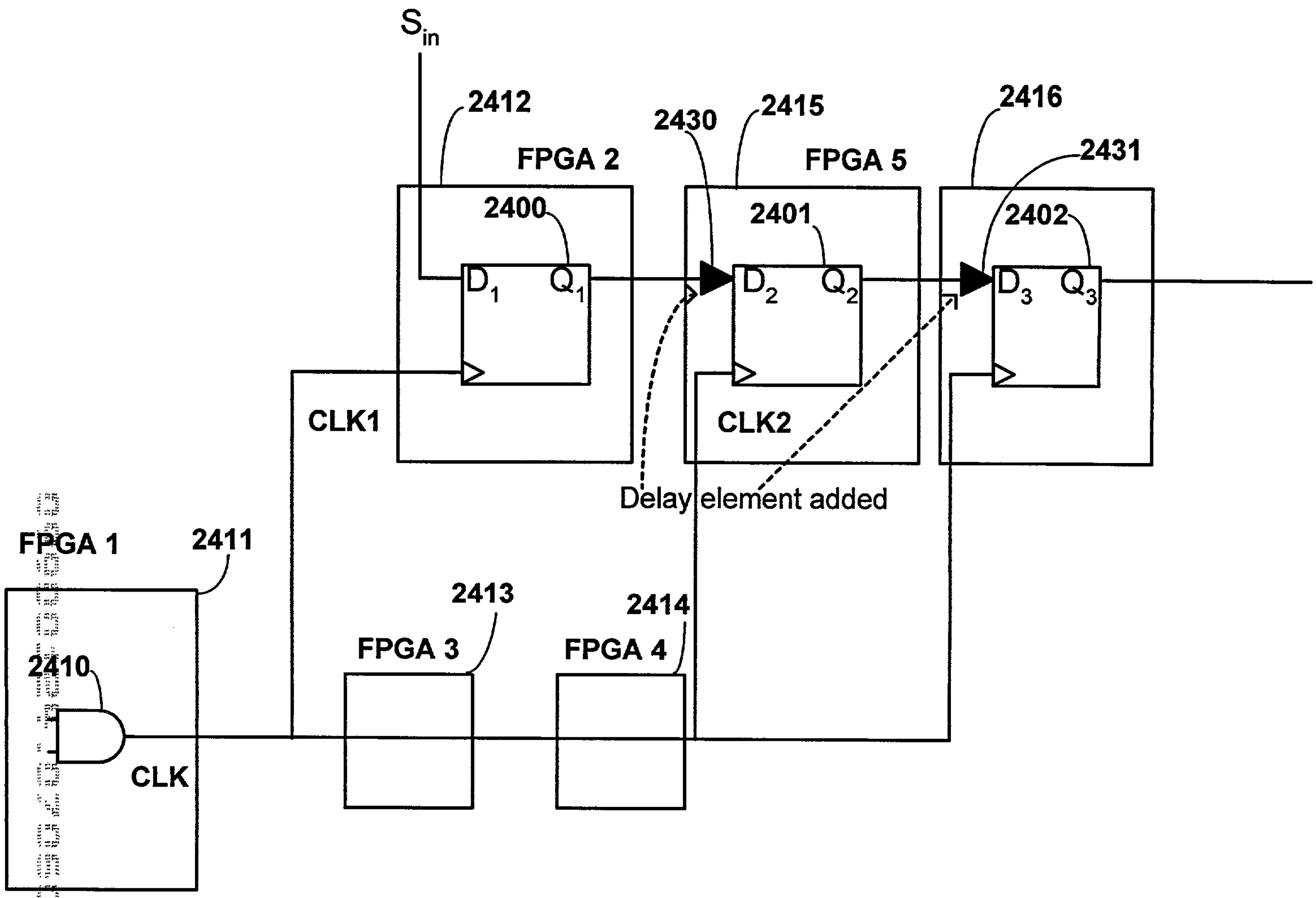


FIG. 77(B)

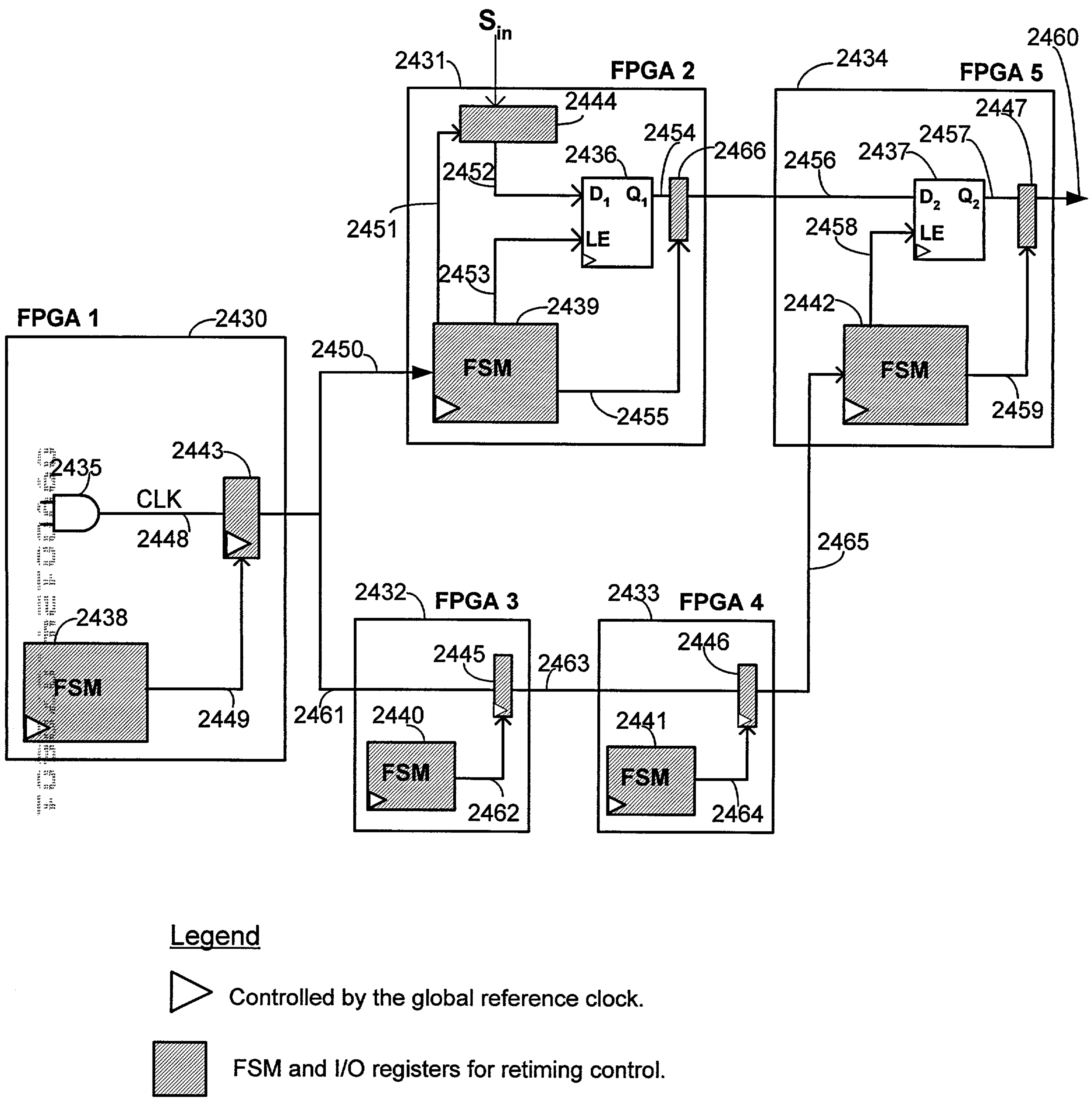
# TIMING ADJUSTMENT BY ADDING DELAY



(Prior Art)

FIG. 78

# GLOBAL RETIMING



(Prior Art)

FIG. 79

# TIGF LATCH

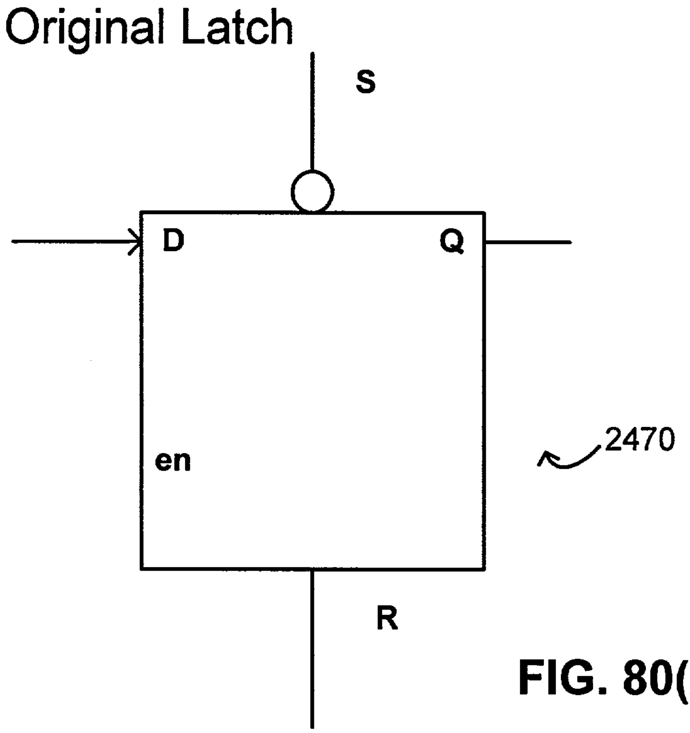


FIG. 80(A)

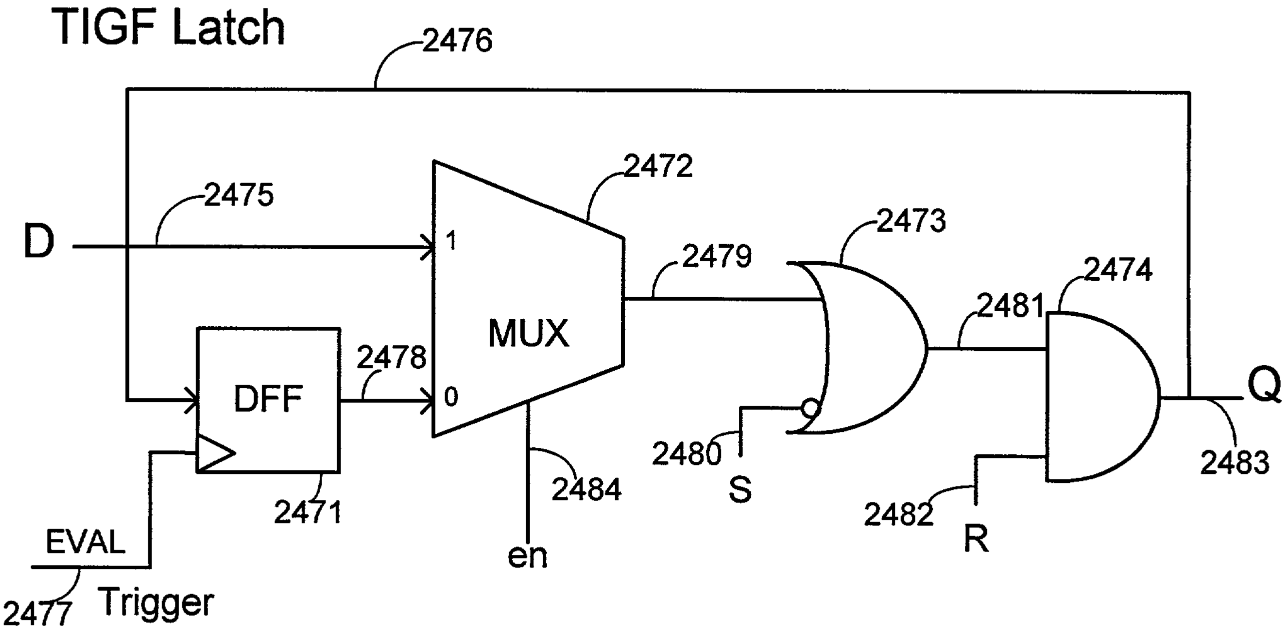


FIG. 80(B)

# TIGF DFF

Original DFF

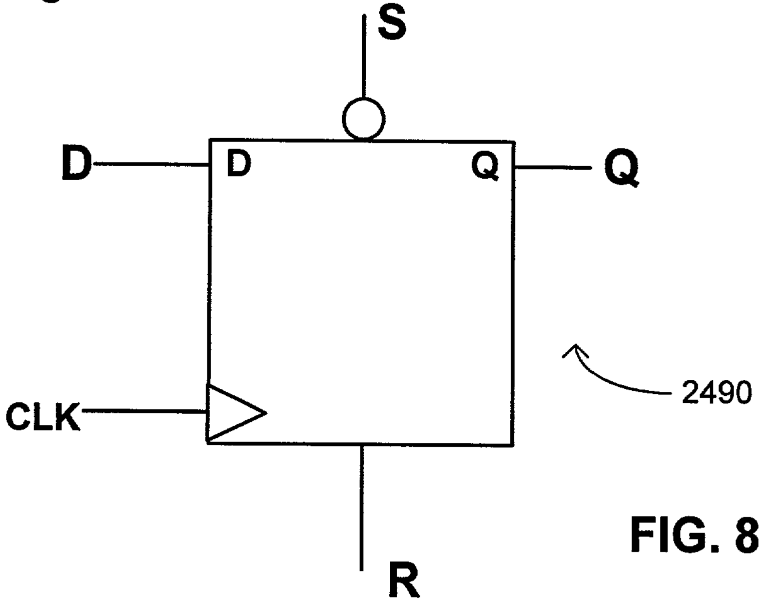


FIG. 81(A)

TIGF DFF and Edge Detector

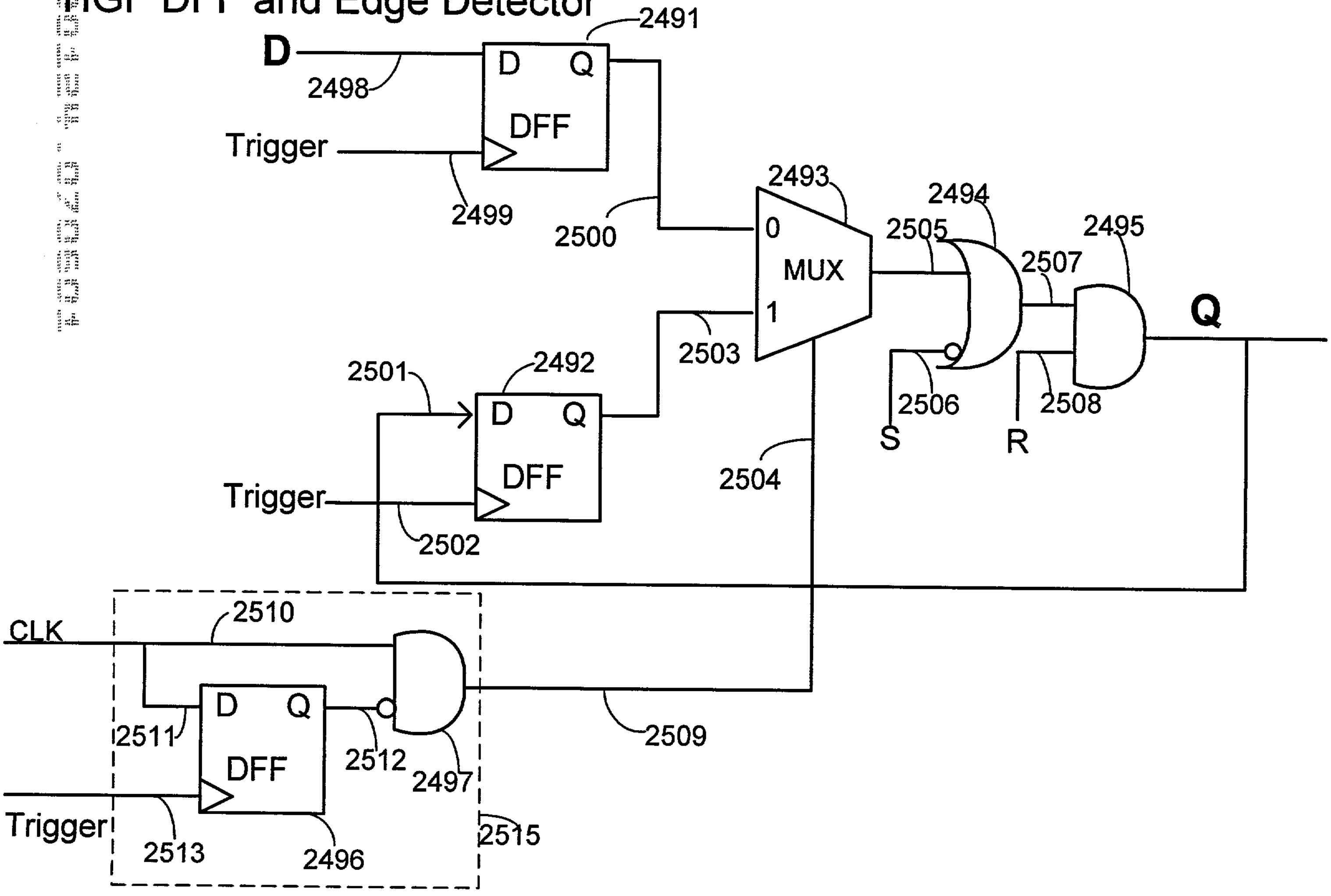


FIG. 81(B)

GLOBAL TRIGGER SIGNAL

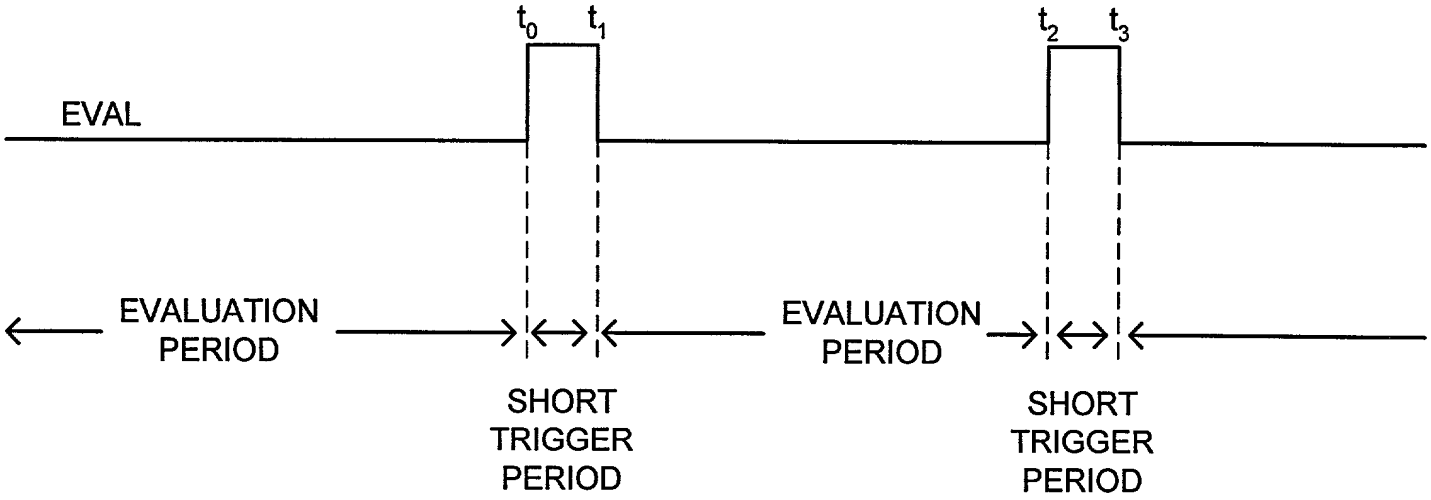


FIG. 82

# RCC System

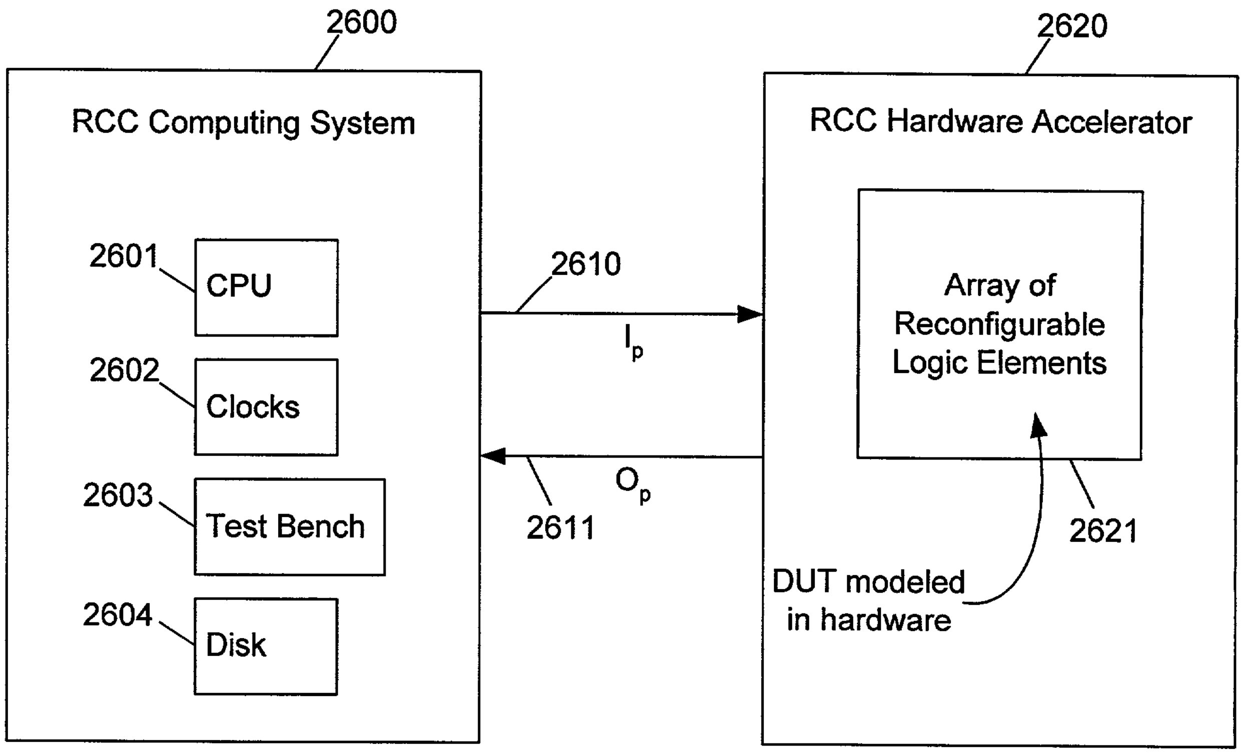


FIG. 83

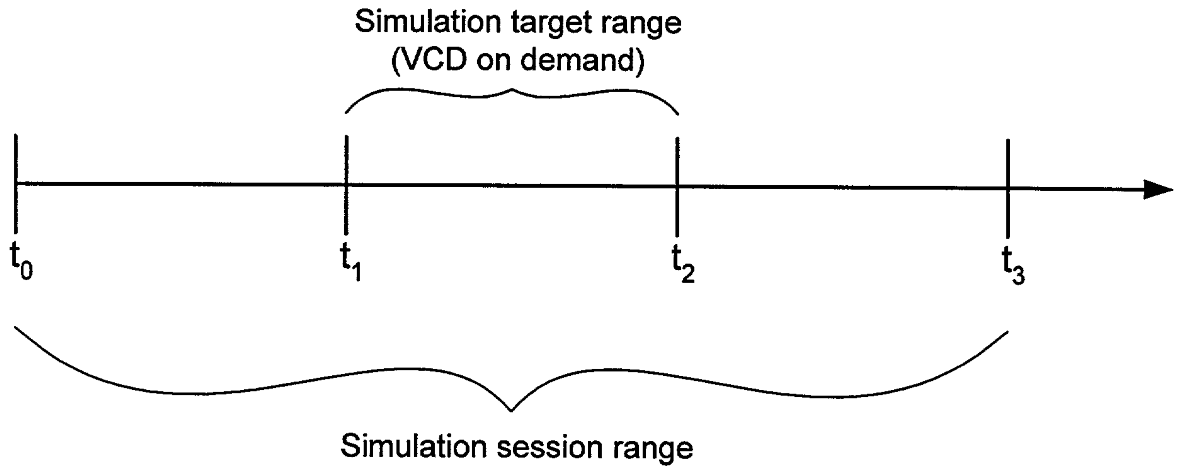

Simulation target range
(VCD on demand)
$t_0$
$t_1$
$t_2$
$t_3$
Simulation session range

FIG. 84

SINGLE-ROW FPGA PER BOARD

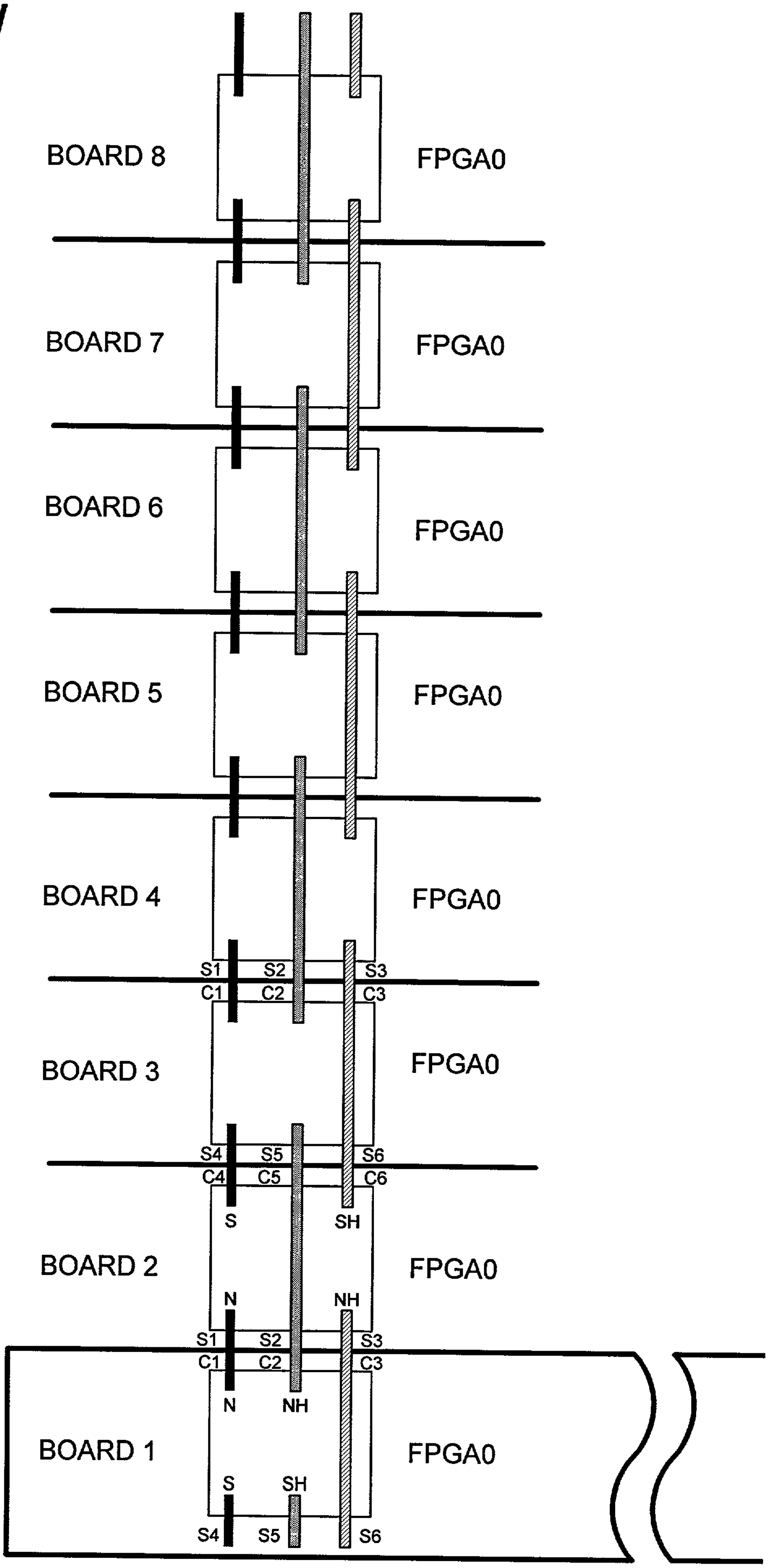


FIG. 85

TWO-ROW FPGA PER BOARD

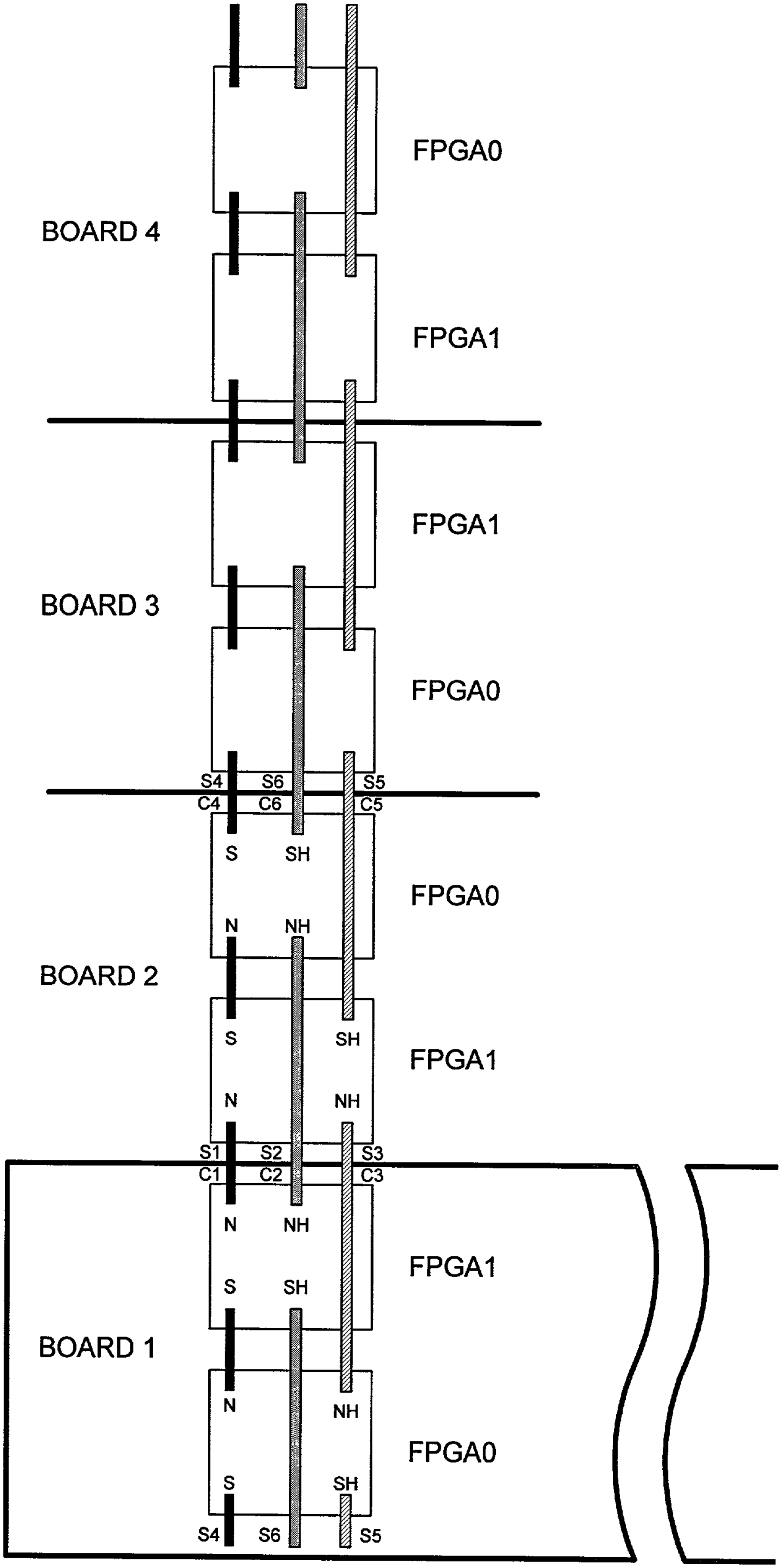


FIG. 86

THREE-ROW
FPGA PER
BOARD

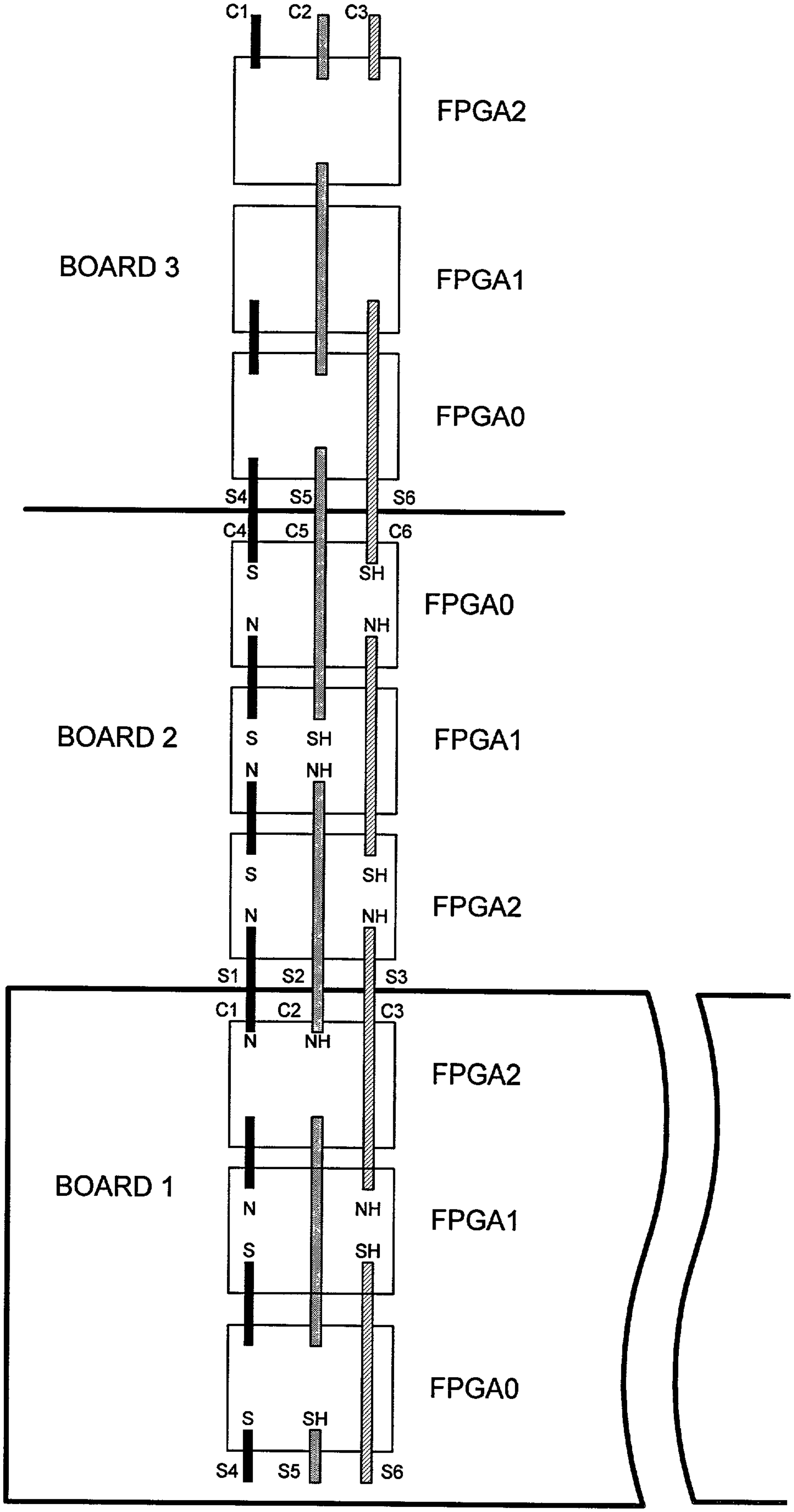


FIG. 87

FOUR-ROW FPGA PER BOARD

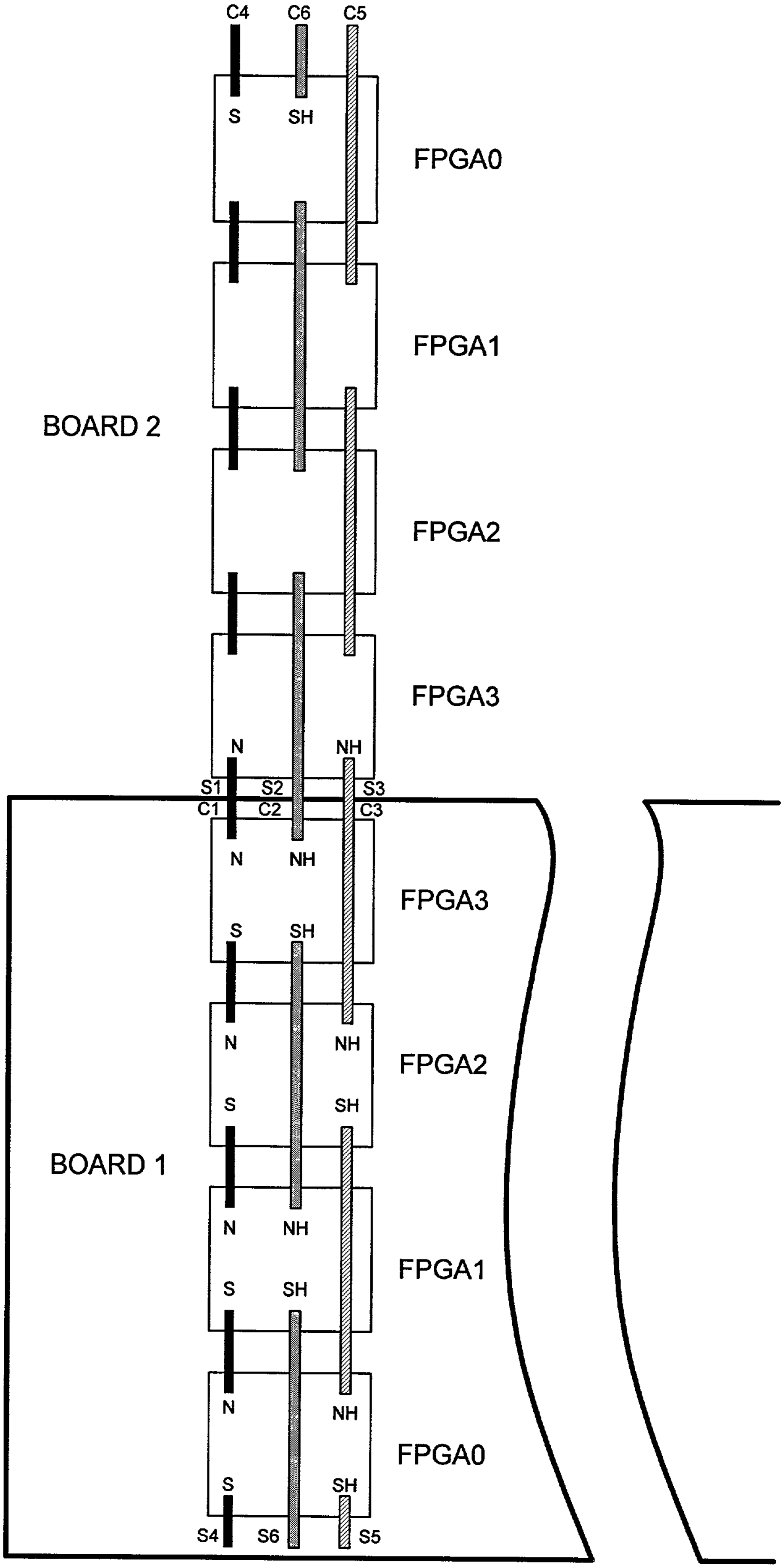


FIG. 88

# INTERCONNECT FOR THREE-ROW PER BOARD

| I/O Signals | Odd Board | Even Board | Common Board |
|---|---|---|---|
| | Connector-Group Pin-position | Connector-Group Pin-position | Connector-Group Pin-position |
| FPGA2_N | C1 | S1 | C1, S1 |
| FPGA2_NH | C2 | S3 | C2, S3 |
| FPGA1_NH | C3 | S2 | C3, S2 |
| FPGA0_S | S4 | C4 | C4, S4 |
| FPGA0_SH | S5 | C6 | C6, S5 |
| FPGA1_SH | S6 | C5 | C5, S6 |

FIG. 89

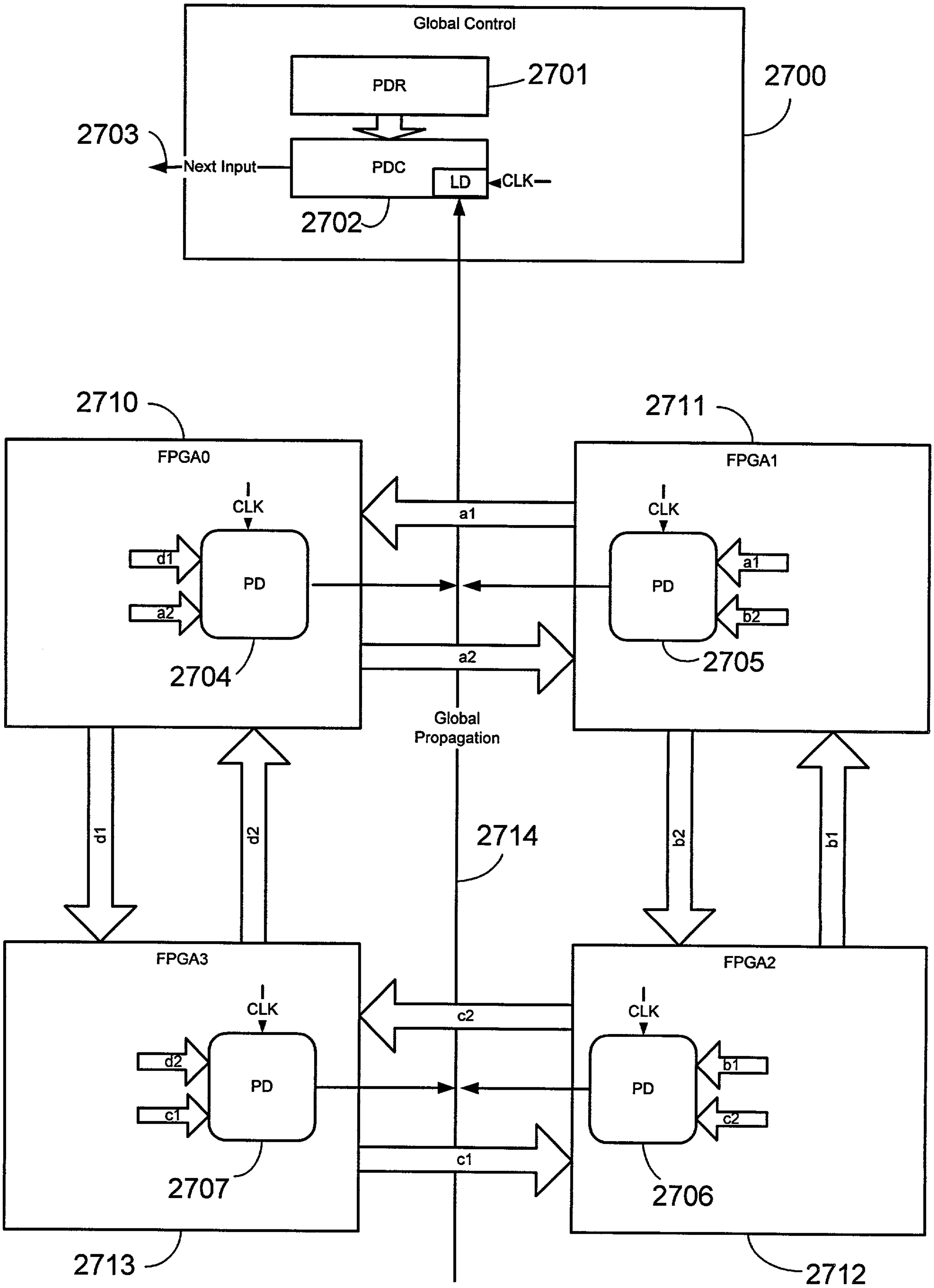
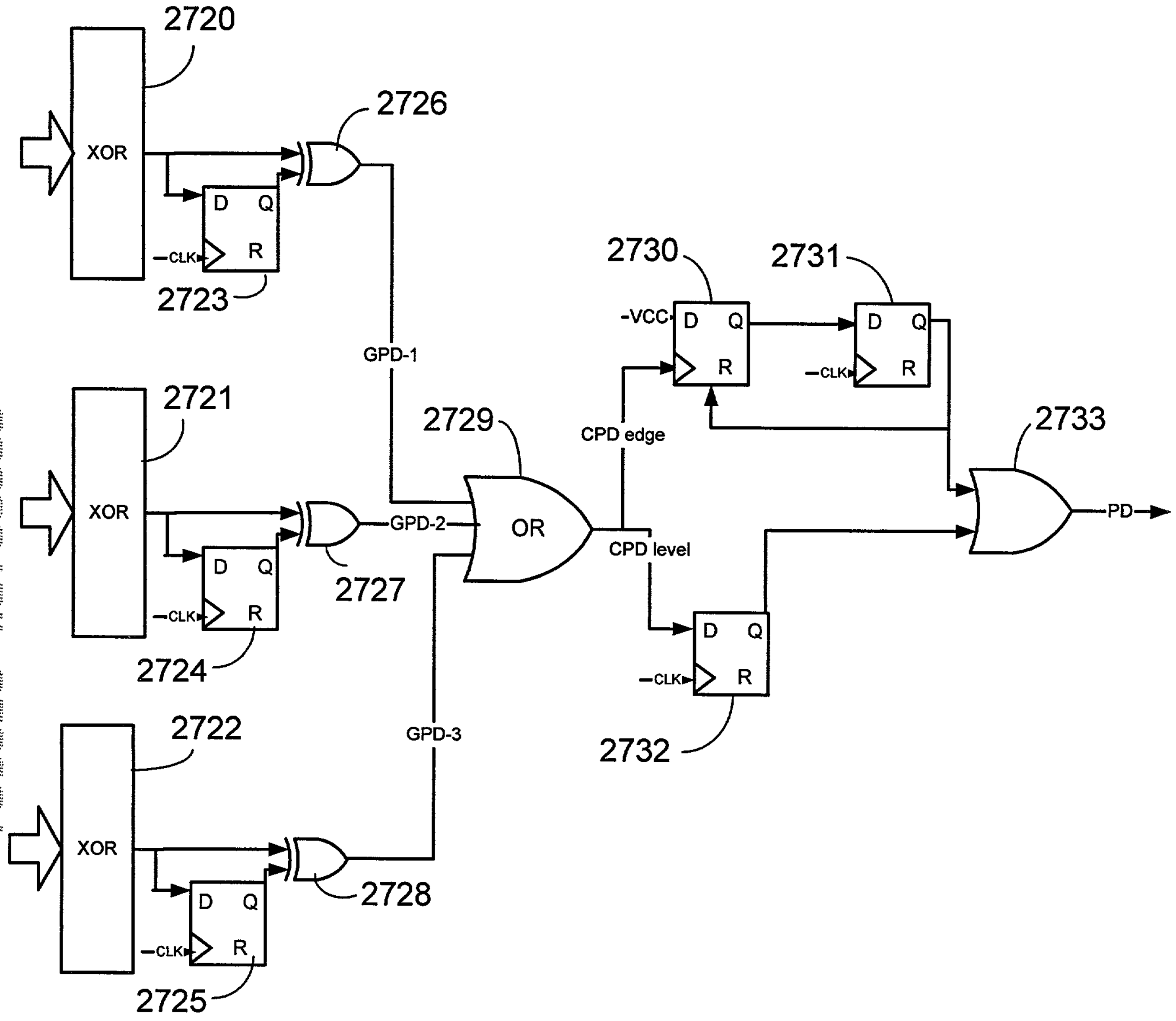

Global Control
PDR
2701
2700
2703
Next Input
PDC
LD
CLK
2702
2710
FPGA0
CLK
d1
a2
PD
2704
2711
FPGA1
CLK
a1
b2
PD
2705
a1
a2
Global Propagation
d1
d2
2714
b2
b1
FPGA3
CLK
d2
c1
PD
2707
FPGA2
CLK
b1
c2
PD
2706
c2
c1
2713
2712

FIG. 90


2720
XOR
2726
D
Q
CLK
R
2723
GPD-1
2730
2731
-VCC
D
Q
R
D
Q
CLK
R
2721
XOR
2729
CPD edge
2733
D
Q
GPD-2
OR
CPD level
PD
CLK
R
2727
D
Q
2724
CLK
R
2732
2722
GPD-3
XOR
D
Q
CLK
R
2728
2725

# FIG. 91

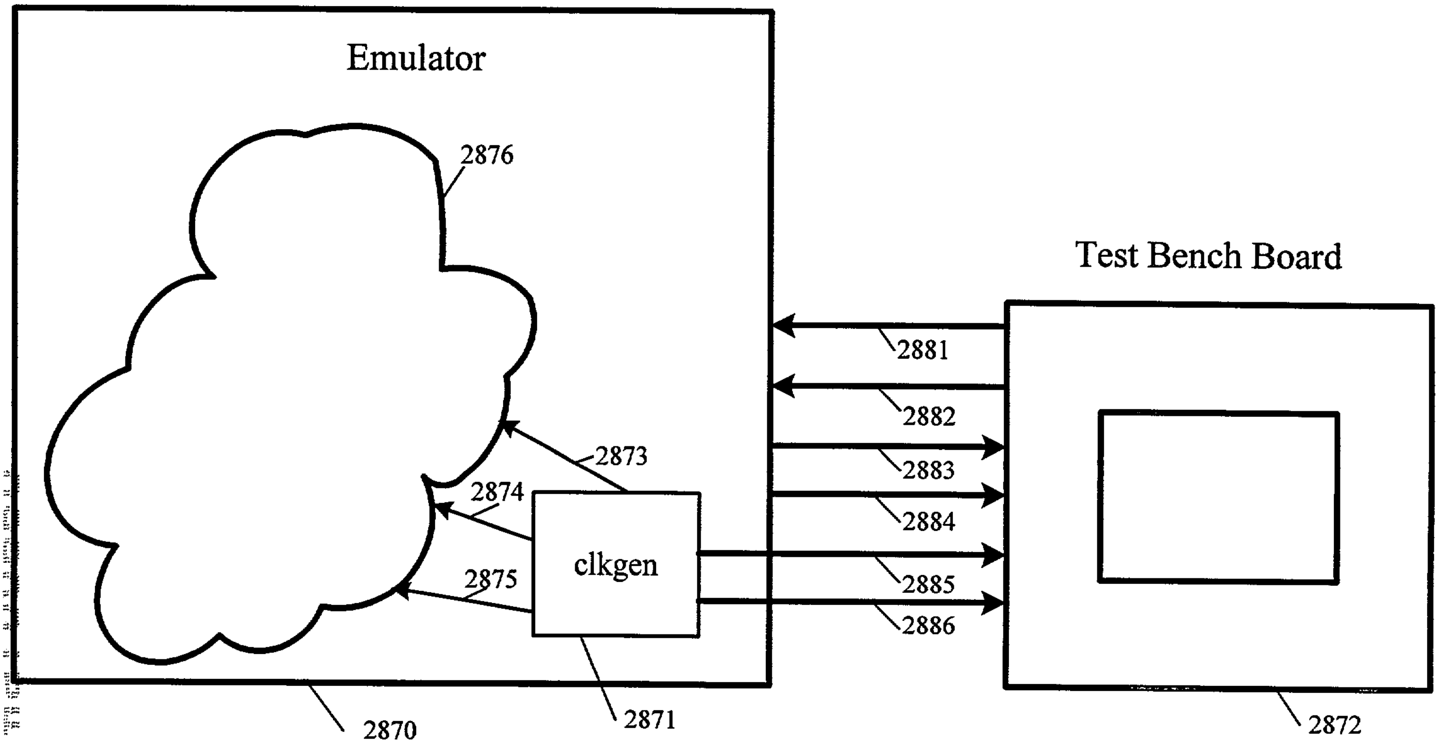

Emulator
2876
Test Bench Board
2881
2882
2873
2883
2874
2884
clkgen
2875
2885
2886
2870
2871
2872

FIG. 92

# Clock Specification

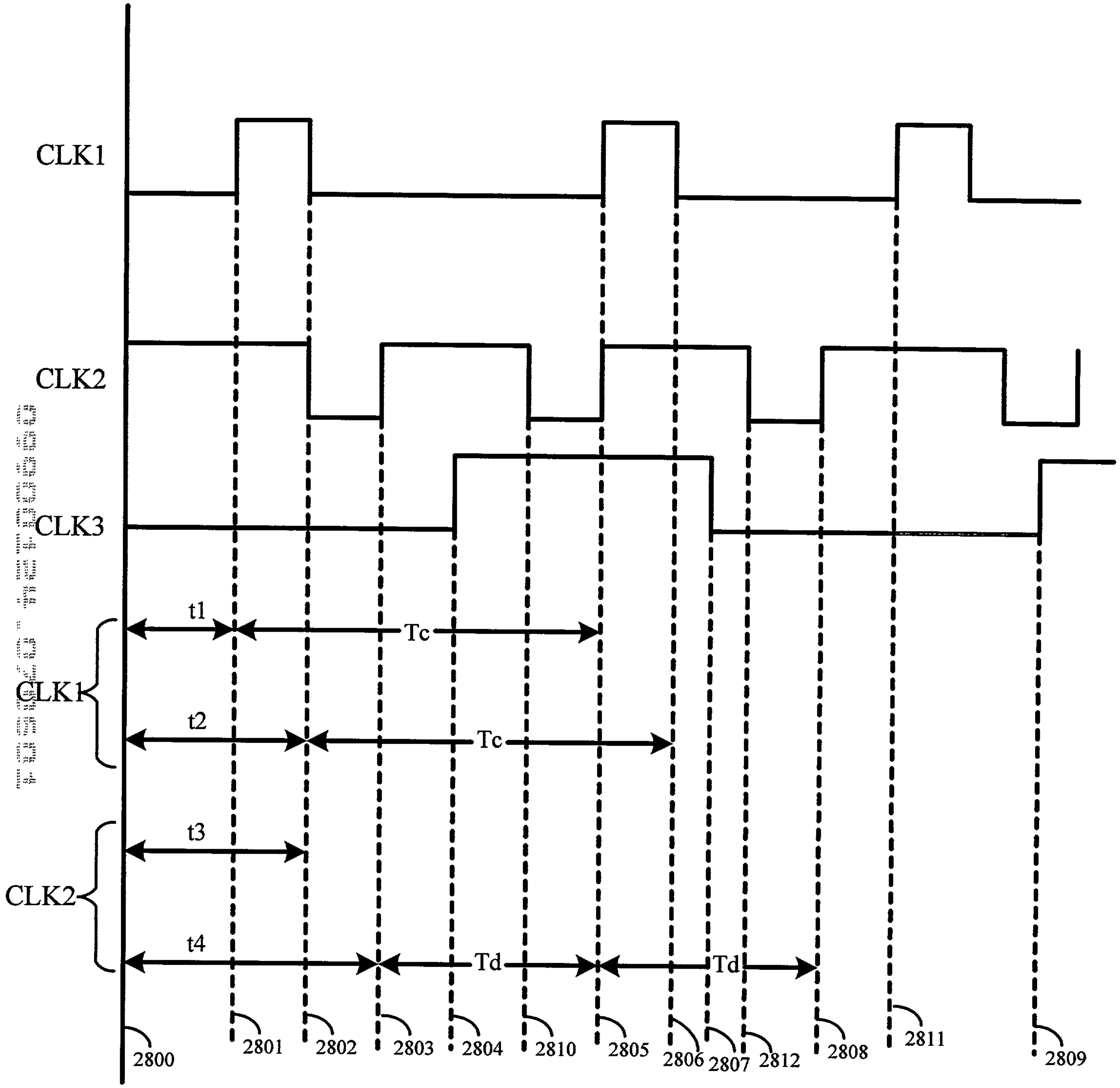


FIG. 93

# Clock Generation Scheduler w/ Slices

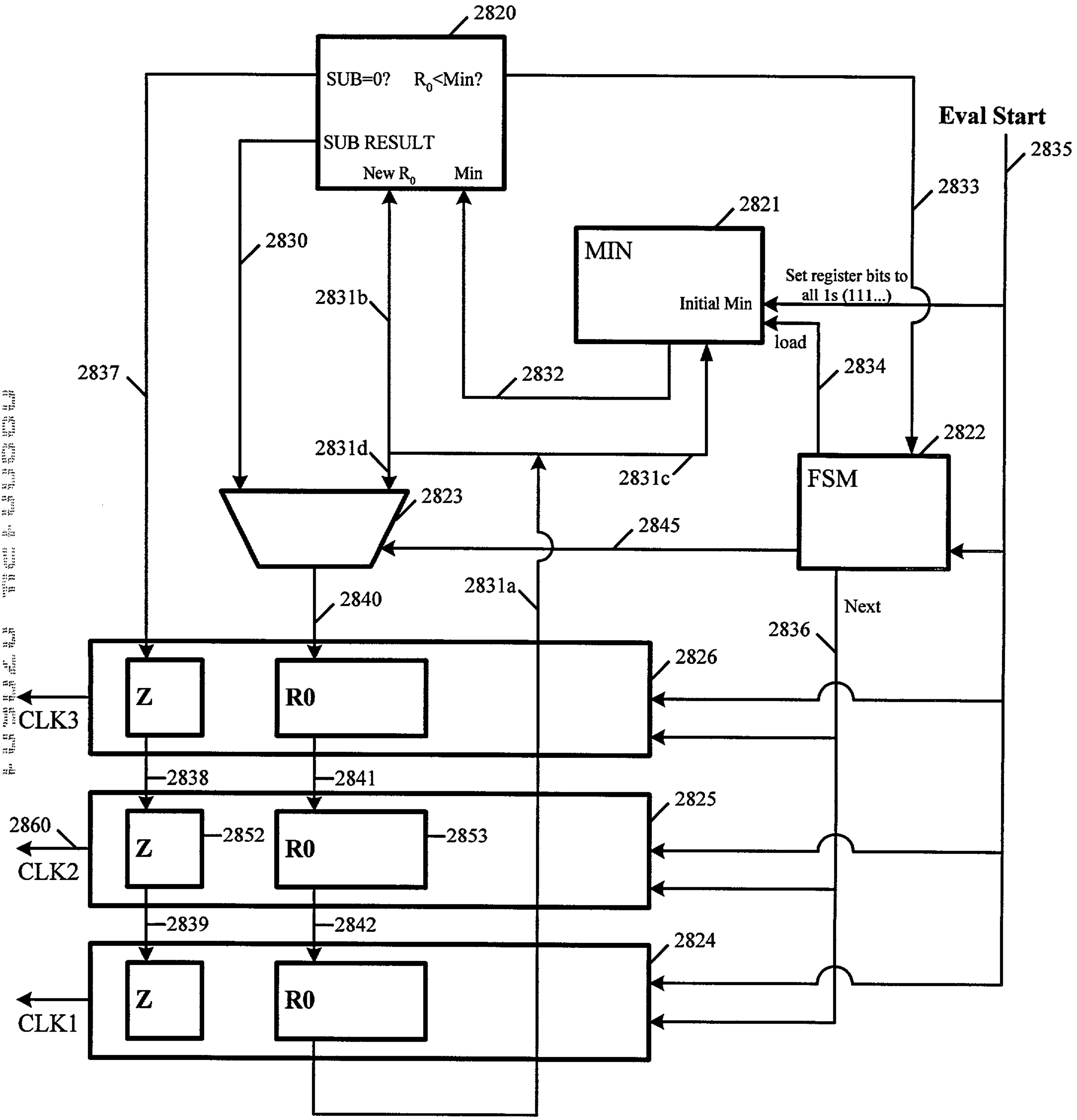


FIG. 94

# Clock Generation Slice

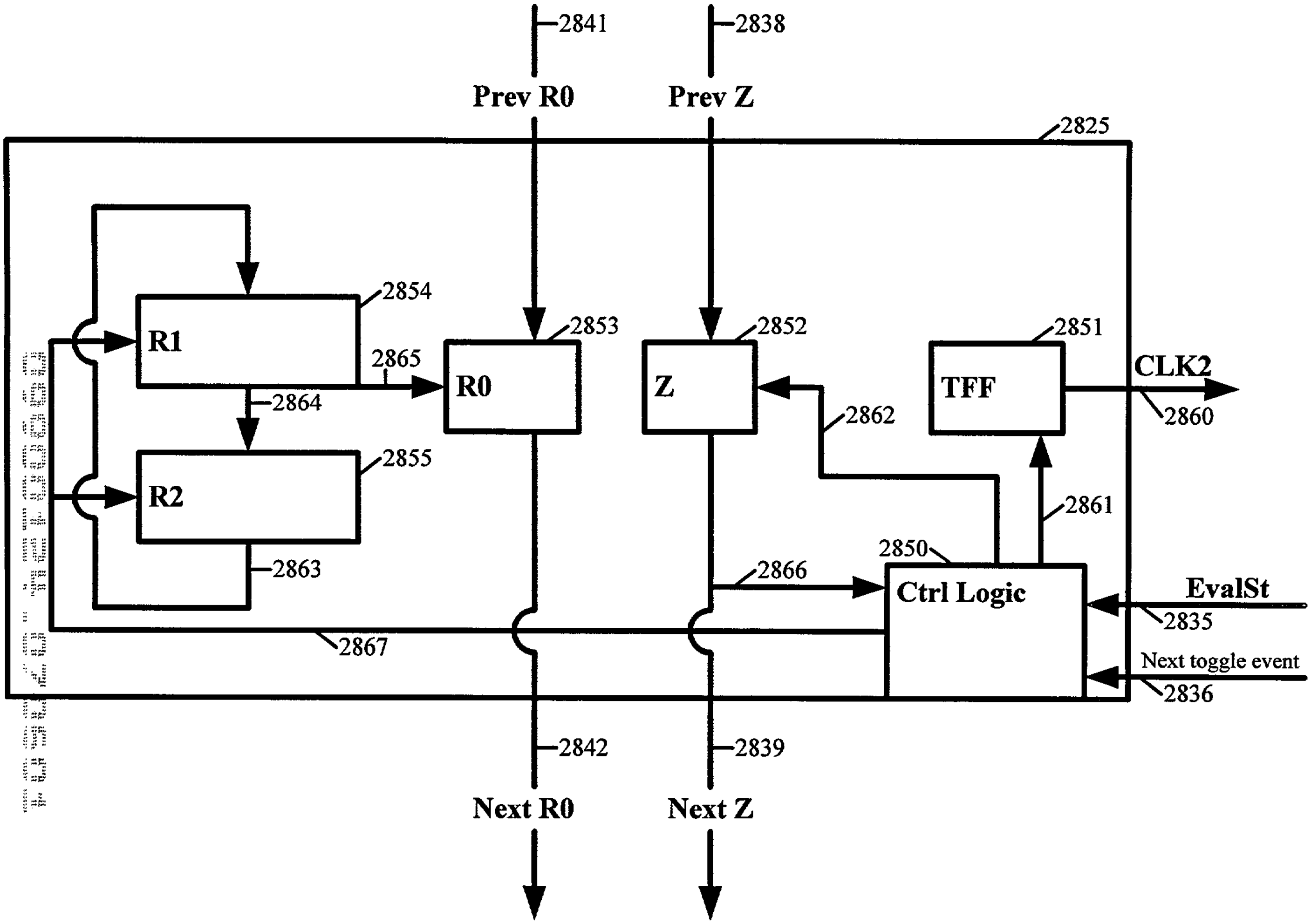


FIG. 95

# Clock Generation Scheduler and Slices

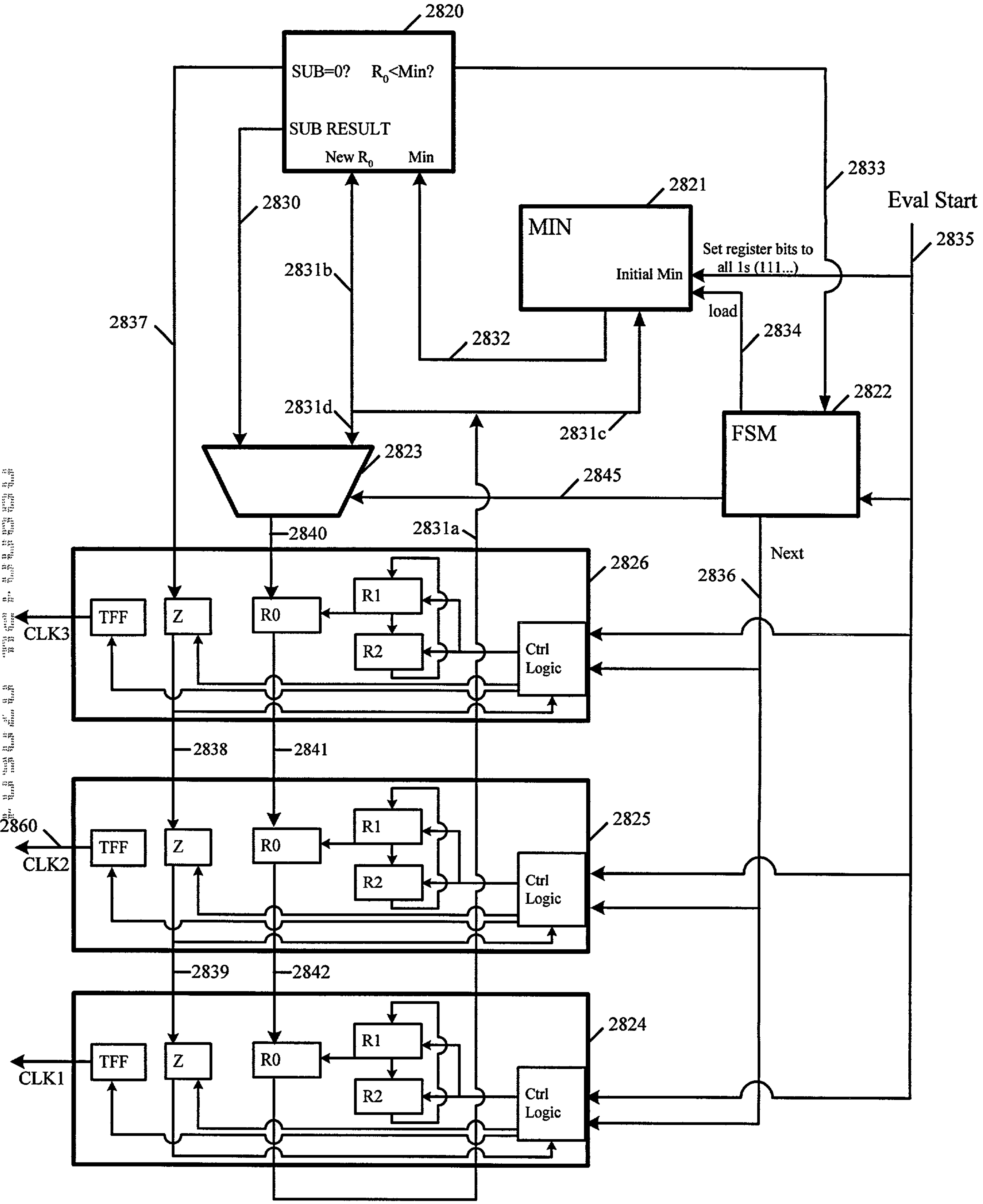


FIG. 96

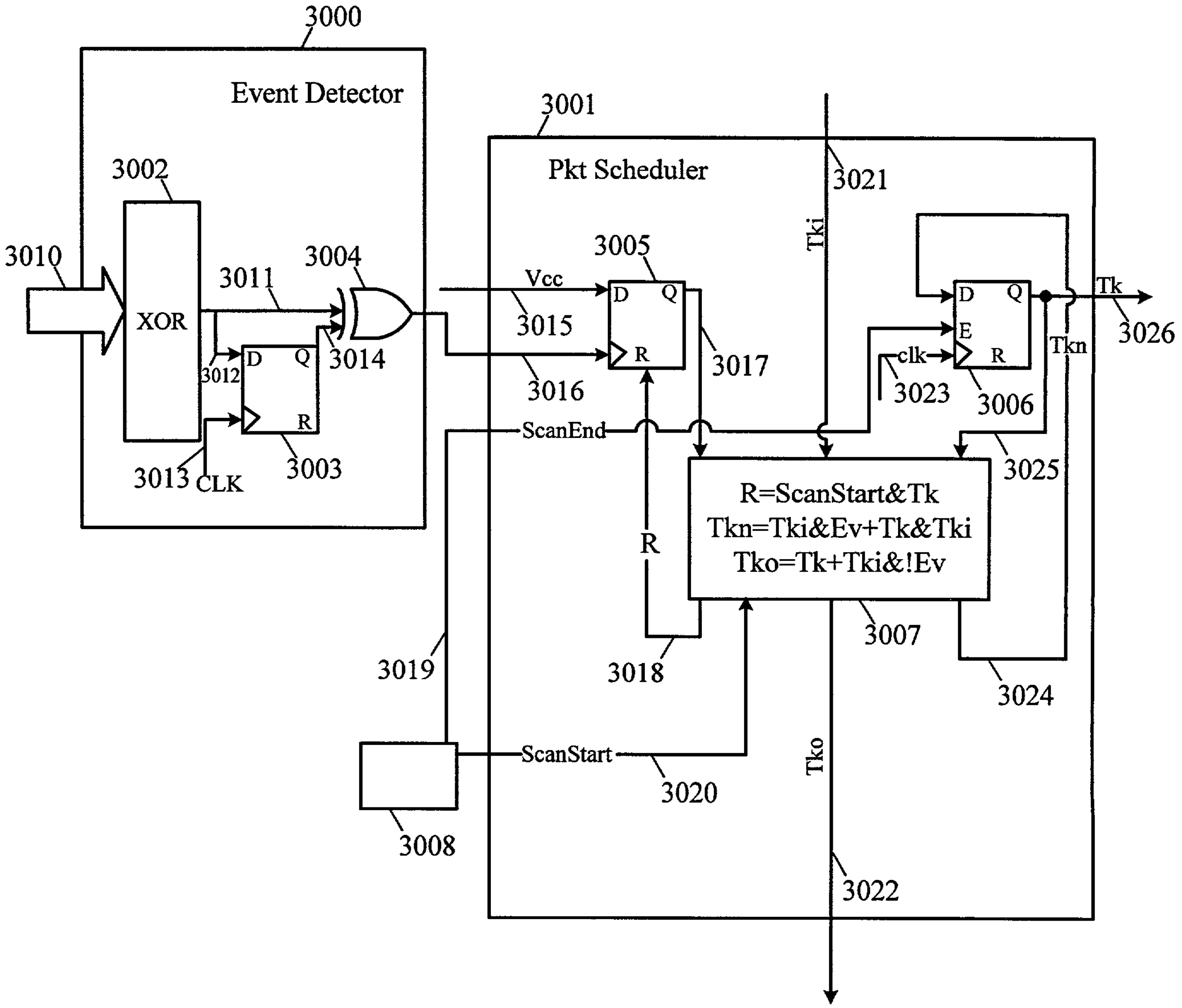

3000
Event Detector
3001
Pkt Scheduler
3002
XOR
3010
3011
3004
3012
D
Q
3014
3003
R
3013
CLK
3005
Vcc
D
Q
3015
R
3016
3017
3021
Tki
D
Q
E
clk
R
Tk
3026
Tkn
3023
3006
ScanEnd
3025
R=ScanStart&Tk
Tkn=Tki&Ev+Tk&Tki
Tko=Tk+Tki&!Ev
R
3007
3019
3018
3024
ScanStart
3020
Tko
3008
3022

FIG. 97

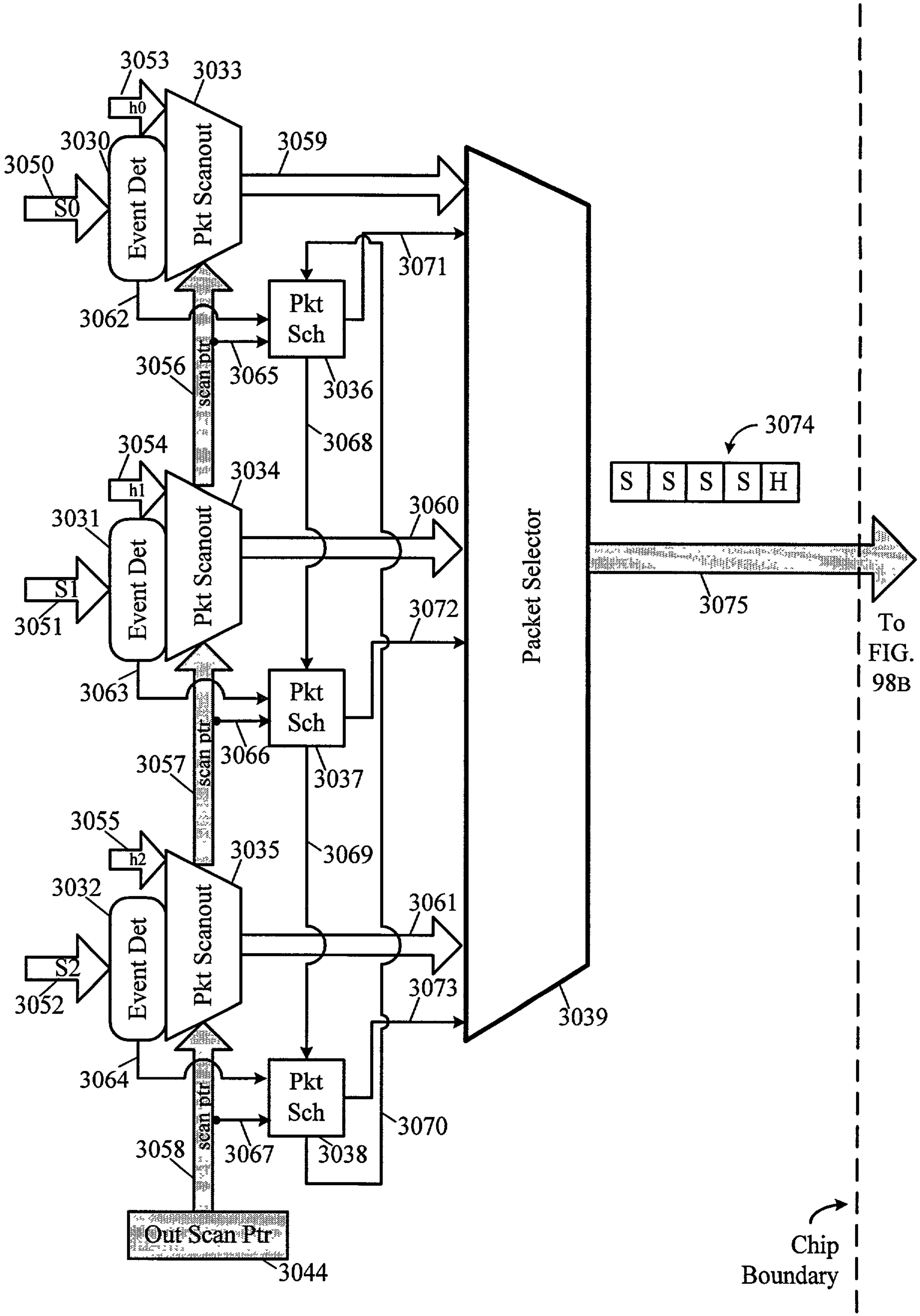

3053
3033
h0
3030
3050
S0
Event Det
Pkt Scanout
3059
3071
3062
Pkt Sch
3056
scan ptr
3065
3036
3068
3074
S
S
S
S
H
3054
3034
h1
3031
S1
3051
Event Det
Pkt Scanout
3060
Packet Selector
3075
To FIG. 98B
3072
3063
Pkt Sch
3066
3057
scan ptr
3037
3055
h2
3035
3069
3032
S2
3052
Event Det
Pkt Scanout
3061
3073
3039
3064
Pkt Sch
scan ptr
3067
3038
3070
3058
Out Scan Ptr
3044
Chip Boundary

FIG. 98A

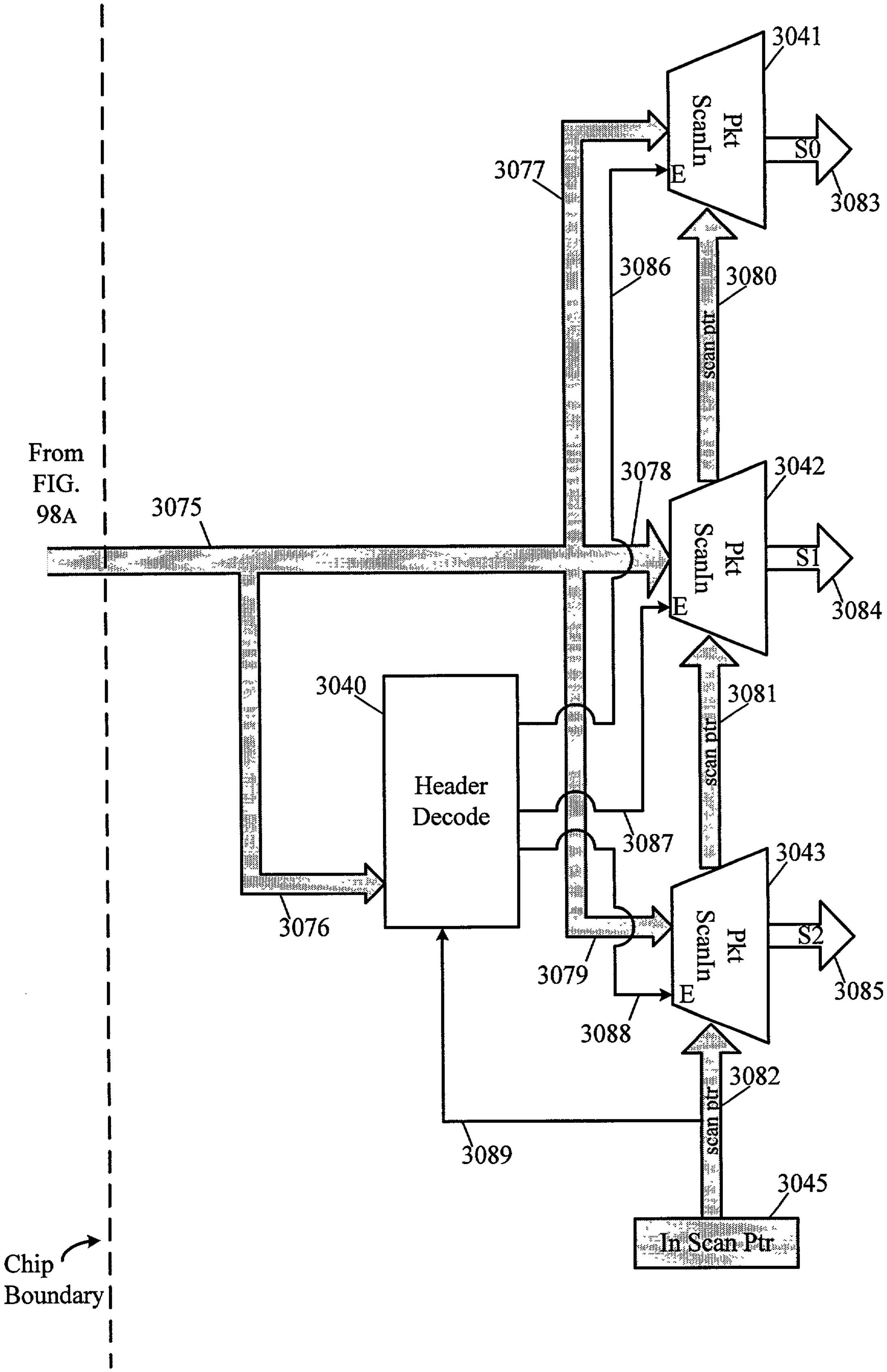

3041
Pkt ScanIn
E
S0
3083
3077
3086
3080
scan ptr
From FIG. 98A
3075
3078
3042
Pkt ScanIn
E
S1
3084
3040
Header Decode
3081
scan ptr
3087
3043
Pkt ScanIn
E
S2
3085
3076
3079
3088
3082
scan ptr
3089
3045
In Scan Ptr
Chip Boundary

FIG. 98B